

# Media

## IN THIS APPENDIX

Getting Linux distributions

Creating a bootable CD or DVD

**U**nless you bought a computer with Linux preinstalled or had someone install it for you, you need to find a way to get a Linux distribution and then either install or run it live on your computer. Fortunately, Linux distributions are widely available and come in a variety of forms.

In this appendix, you learn how to do the following:

- Get a few different Linux distributions
- Create a bootable disk to install your distribution
- Boot Linux from a USB drive

To use this book effectively, you should have a Linux distribution in front of you to work on. It's important to be able to experience Linux as you read. So, try the examples and do the exercises.

Linux distributions are most commonly available from the websites of the organizations that produce them. The following sections describe websites associated with Linux distributions that offer ISO images you can download.

### NOTE

An ISO is a disk image that is formatted in the ISO 9660 filesystem format, a format that is commonly used with CD and DVD images. Because this is a well-known format, it is readable by Windows, Mac, and Linux systems.

An ISO image can be used to create a bootable USB flash drive, CD, or DVD medium, depending on the size of the image. An ISO image in your filesystem can be mounted in Linux in loopback mode, so you can view or copy its contents.

When an ISO image contains a Linux Live CD or installation image, the images are bootable. This means that instead of starting up an operating system, such as Windows or Linux, from the computer's hard drive, you can tell your computer to boot from the CD or DVD instead. This enables you to run a totally different operating system than is installed on your hard drive without changing or damaging the data on that drive.

## Getting Fedora

### NOTE

I recommend downloading the Fedora Workstation Live Image to use along with this book because most of the book works with that distribution. You can run it live without committing to overwriting your computer's hard disk until you feel comfortable enough to install it permanently.

To test the examples in this book, I used Fedora 30 and 31, 64-bit Fedora Workstation images, which you can get from GetFedora.org (<https://getfedora.org/en/workstation/download>). If you have a 64-bit machine, you must use the 64-bit ISO.

Later versions of Fedora that come with a GNOME desktop should work as well. Here's a link to the exact ISO used for the Fedora 31 Workstation:

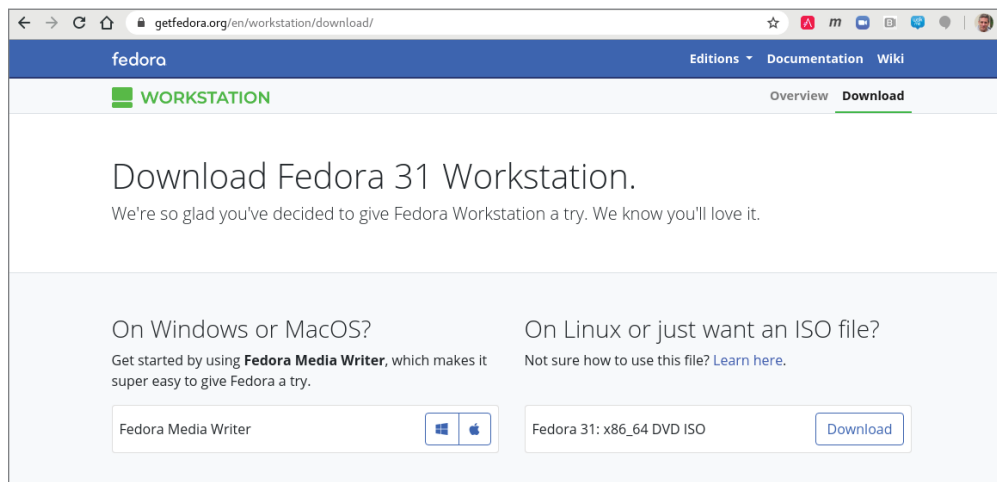
```
https://download.fedoraproject.org/pub/fedora/linux/releases/31/Workstation/x86\_64/iso/Fedora-Workstation-Live-x86\_64-31-1.9.iso
```

Keep in mind that the latest Fedora Workstation ISO image does not fit on a CD, so you must burn it to a DVD or USB flash drive. See the descriptions of CD/DVD burning tools available for Windows, MacOS, and Linux later in this appendix.

Figure A.1 shows an example of the Get Fedora page.

**FIGURE A.1**

Download Fedora ISO images from the Get Fedora page.



Today, the default download is an ISO image of a 64-bit PC-type Fedora Workstation (GNOME) Live DVD. You can boot this image on your computer, and if you choose, you can permanently install it to your computer's hard drive. To download this image, do the following:

1. Select Workstation or Server from [GetFedora.org](http://GetFedora.org). I recommend Workstation to follow along with this book.
2. Select the Download Now button and click the Download button. A pop-up should appear, asking what you want to do with the ISO.
3. Select to save the ISO. Depending on your settings, either you are asked where you want to download it or it simply begins downloading to a default folder (in Linux, it is probably a Downloads folder).
4. If you are prompted for where to put the ISO, select a folder that has enough space to hold it. Remember where this folder is located, because you need to find the ISO when you go to burn it later.

If you need more information about what to do with the downloaded image, there are links to help you on the Fedora page that appears. At the time of this writing, the Learn Here link takes you to descriptions of how to create live installation media. The exact instructions might change as the website is updated.

You have other choices for downloading ISOs from Fedora. From the bottom of the [GetFedora.org](http://GetFedora.org) page, you can download specially configured Fedora ISO images called spins (<https://spins.fedoraproject.org>). Here are some special types of Fedora spins that might interest you:

**KDE desktop spin:** People who prefer the KDE desktop to the GNOME desktop can download the Plasma KDE spin.

**Lightweight desktop spin:** If you are trying Linux on a computer with less memory or processing power, consider Xfce and LXQt spins (representing lightweight desktops of the same name).

**Desktop effects spin:** The MATE-Compiz spin offers more of the other extreme to the lightweight desktops, with desktop effects like wobbly windows and desktops that rotate on a cube.

**Child-friendly desktop spin:** The SOAS desktop is a spin of the Sugar Learning Platform, made to provide a simplified setup and a child-friendly graphical interface. SOAS can be transported on a USB drive and run on any available computer.

## Getting Red Hat Enterprise Linux

Many large corporations, government agencies, and universities use Red Hat Enterprise Linux to run their mission-critical applications. While most of the procedures in this book will run well on Fedora, there are many references to how things are done differently in

Red Hat Enterprise Linux because, when you go to get a job as a Linux system administrator, you will, in most cases, be working with Red Hat Enterprise Linux systems.

Although the source code for Red Hat Enterprise Linux is freely available, the ISOs containing the packages you install (often referred to as the *binaries*) are available only to those who have accounts on the Red Hat customer portal (<https://access.redhat.com>) or through evaluation copies.

If you don't have an account, you can try signing up for a 30-day trial. If either you or your company has an account with Red Hat, you can download the ISOs that you need. Go to the following site and follow the instructions to download a Red Hat Enterprise Linux server ISO or sign up to get an evaluation copy:

<https://access.redhat.com/downloads>.

Red Hat does not offer live versions of Red Hat Enterprise Linux. Instead, you can download installation DVDs that you can install as described in Chapter 9, "Installing Linux," of this book.

### NOTE

If you are unable to obtain a Red Hat Enterprise Linux installation DVD, you can get a similar experience using the CentOS installation DVD. CentOS is not exactly the same as RHEL. However, if you download the CentOS installation DVD for CentOS 8.x from links on the CentOS site (<http://www.centos.org/download/>), the installation procedure is similar to the one described for Red Hat Enterprise Linux in Chapter 9.

## Getting Ubuntu

---

Many people new to Linux begin by downloading and installing Ubuntu. Ubuntu has a huge fan base and many active contributors. If you have problems with Ubuntu, there are large, active forums where many people are willing to help you overcome problems.

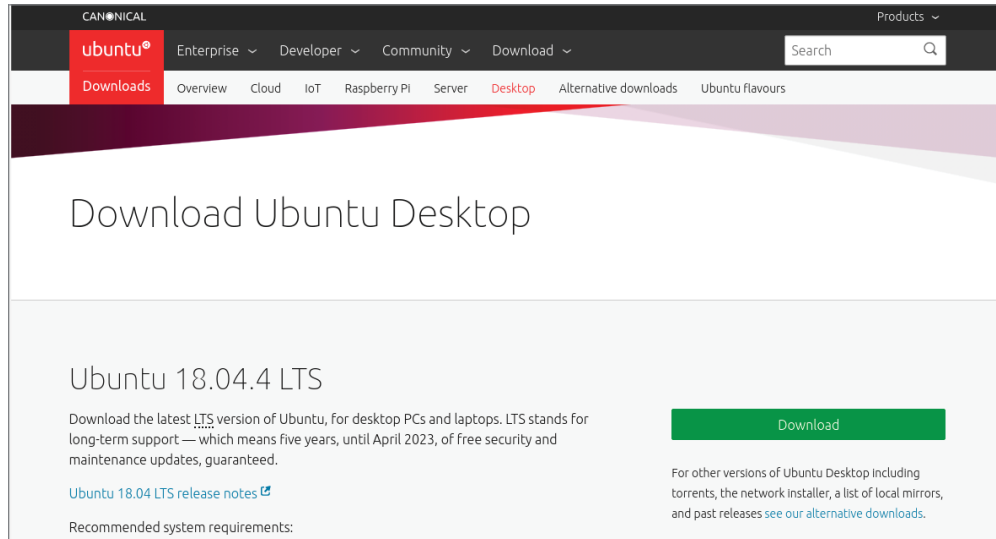
If you already have an Ubuntu system installed, you can follow along with most of this book. You can get Ubuntu with a GNOME desktop, and its default dash shell is similar to bash (or you can switch to bash in Ubuntu to match the shell examples in this book). Although most of the examples of this book focus on Fedora and RHEL, I have added many more references to Ubuntu throughout the book in this edition.

To get Ubuntu, you can download a Live ISO image or installation medium from the Download Ubuntu page: <http://www.ubuntu.com/download/ubuntu>.

Figure A.2 shows an example of the Download Ubuntu Desktop page.

**FIGURE A.2**

Download Ubuntu Live ISO images, or choose an alternative download.



As with Fedora, the easiest way to download Ubuntu is to select the 64-bit Ubuntu Live image, download it, and burn it. Here's how to do that from the Download Ubuntu page:

1. Click the Download button. By default, this downloads the most recent 64-bit Ubuntu desktop Live ISO image.
2. Either you are asked where you want to download the ISO image, or it simply begins downloading to a default folder.
3. If you are asked where to put the ISO, select a folder that has enough space to hold the ISO. Remember where this folder is located because you need to find the ISO when you go to burn it later.

After the download is complete, burn the ISO image to a DVD using procedures described in the section "Creating Linux CDs and DVDs".

Other types of Ubuntu installation media are also available. To find other Ubuntu media, go to the Alternative Downloads page (<http://www.ubuntu.com/download/alternative-downloads>). From this site, you can get media that contains a variety of desktop and server installs.

## Booting Linux from a USB Drive

Instead of burning ISO images to a CD or DVD, you can put your Linux system on a USB drive. USB drives offer the advantage of being writable as well as readable, so you can

save your content between sessions. Most modern computers can boot from a USB drive, although you may have to interrupt the boot process to tell the BIOS to boot from a USB drive instead of hard drive or CD/DVD drive.

You can find procedures for putting Fedora and Ubuntu on a USB drive in the following locations:

**Fedora on a USB drive:** Using a tool called Live USB Creator, you can install a Fedora ISO image to a USB drive in either Windows or Linux. To run Fedora from that drive, insert it into a USB port on your computer, reboot the computer, interrupt the BIOS as it is booting (possibly F12), and select to boot from a USB drive. The procedure for using Live USB creator is located at

<https://docs.fedoraproject.org/en-US/quick-docs/creating-and-using-a-live-installation-image/index.html>

**Ubuntu on a USB drive:** Ubuntu has procedures for creating a bootable USB drive with Ubuntu on it that work from Windows, MacOS, or Linux. To find out how to do this, go to the Ubuntu Download page, and under “Easy ways to switch to Ubuntu,” look for the appropriate “How to create a bootable USB stick...” procedure for Ubuntu, Windows, or MacOS:

<https://ubuntu.com/tutorials/tutorial-create-a-usb-stick-on-ubuntu#1-overview>

## Creating Linux CDs and DVDs

---

After you have downloaded a Linux CD or DVD image, you can use several tools to create bootable CDs or DVDs for either installing or just running Linux live from those media. Before you begin, you must have the following:

**DVD or CD ISO images:** Download the ISO images to your computer that represent the physical DVD or CD you will ultimately burn. Today, most Linux ISO images are too big to fit on a CD (including those for RHEL, Fedora, and Ubuntu).

**Blank DVDs/CDs:** You need blank DVDs or CDs to burn the images to. CDs hold up to about 700MB; DVDs hold up to about 4.7GB (single layer).

**CD/DVD burner:** You need a drive that is capable of burning CDs or DVDs, depending on which you are burning. Not all CD/DVD drives can burn DVDs (especially older ones). So, you may need to find a computer with a drive that has that capability.

The following sections describe how to burn bootable CDs and DVDs from Windows, MacOS, and Linux systems.

### Burning CDs/DVDs in Windows

If you have downloaded your Linux ISO image to a Windows system, you can burn that image to CD or DVD in different ways. Here are some examples:

**Windows:** In the latest Windows releases, the function of burning ISO images to CD or DVD is built into Windows. After an ISO image is downloaded, simply insert the

appropriate CD or DVD into your computer's drive (assuming the drive is writeable), right-click the ISO image icon from the folder to which you downloaded it, and select Burn Disc Image. When the Windows Disc Image Burner window appears, select Burn to burn the image.

**Roxio Creator:** This third-party Windows application contains many features for ripping and burning CDs and DVDs. You can read about the product here: <http://www.roxio.com/en/products/creator/>.

**Nero CD/DVD Burning ROM:** Nero is another popular CD/DVD burning software product for Windows systems. You can find out more about Nero here: <http://www.nero.com>.

## Burning CDs/DVDs on a MacOS system

Like Windows, MacOS has CD/DVD burning software built into the operating system. To burn an ISO image to disk on a MacOS system, follow these steps:

1. Download the ISO image you want on your MacOS system. An icon representing the ISO should appear on your desktop.
2. Insert a blank CD or DVD into your CD/DVD burner, as appropriate for the size of the image.
3. Right-click the icon representing the Linux ISO that you just downloaded and select Burn "Linux" to Disk. A pop-up window appears, asking if you are sure you want to burn the image.
4. Fill in the name that you want to give the ISO and the write speed and then select Burn. The image begins burning to disk.
5. After the image has been burned, eject the disk; you are ready to boot the CD or DVD on an appropriate computer.

## Burning CDs/DVDs in Linux

Linux has both graphical and command-line tools for burning CD and DVD images to physical media. Examples in this section show how to use K3b from the desktop or `cdrecord` (or `wodim`) to burn ISO images to CD or DVD. If they are not installed, you can install either one as follows:

### For Fedora or RHEL

```
# yum install k3b
# yum install wodim
```

### For Debian or Ubuntu

```
# apt-get install k3b
# apt-get install wodim
```

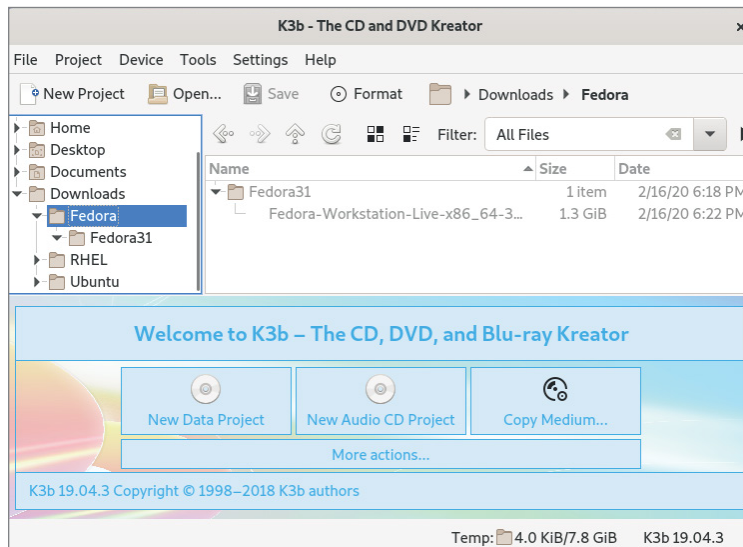
### Burning CDs or DVDs from a Linux desktop

Here's how to create bootable Linux CDs or DVDs from a running Linux system (such as Fedora) using K3b. K3b comes with the KDE desktop but runs on the GNOME desktop as well.

1. Download the ISO images that you want to your computer's hard drive. (A CD image is under about 700MB in size. Single-layer DVD images are under 4.7GB.)
2. Open a CD/DVD burning application. For this procedure, I recommend K3b CD and DVD Kreator (<http://www.k3b.org>). In Fedora, select Activities and type **K3b** (or type **k3b** from a Terminal window). The “K3b – The CD and DVD Kreator” window appears.
3. From the K3b window, select Tools ⇄ Burn Image to burn a CD or DVD ISO Image. You are asked to choose an image file.
4. Browse to the image that you just downloaded or copied to hard drive and select it. After you select the image that you want, the Burn Image window appears, as does a checksum on the image. Figure A.3 shows the K3b window ready to select an image of Fedora.

**FIGURE A.3**

Use K3b to burn your Linux CDs or DVDs.



5. Insert a blank CD or DVD into the CD/DVD drive, which may be a combination CD/DVD drive. (If a CD/DVD Creator window pops up, you can close it.)



6. Check the settings in the Burn Image window (often, the defaults are fine, but you may want to slow down the speed if you get some bad burns). You can also select the Simulate check box to test the burn before actually writing to the CD/DVD. Click Start to continue.
7. When the CD/DVD is finished burning, eject it (or it may eject automatically) and mark it appropriately (information such as the distribution name, version number, date, and name of the ISO image).

Now you're ready to begin installing (or booting) the Linux distribution you just burned.

### **Burning CDs or DVDs from a Linux command line**

If you have no GUI, or you don't mind working from the shell, you can use the `cdrecord` command to burn the ISOs. With a blank CD or DVD inserted and the ISO image you want to burn in the current directory, you can use the following simple command line for burning a CD image to CD or DVD using `cdrecord`:

```
# cdrecord -v whatever.iso
```

See the `cdrecord` man page (`man cdrecord`) for other options available with the `cdrecord` command.

# Exercise Answers

**T**his appendix provides answers to each of the chapter exercises. There are many ways to accomplish tasks in Linux. Suggested answers are provided herein.

Some of the exercises require that you modify system files that could change the basic functioning of your system, or even make your system unbootable. Therefore, I recommend that you do the exercises on a Linux system that you are free to modify and erase if something should go wrong. Using virtual machines, that you can discard when you are done, is an excellent option.

## Chapter 1: Starting with Linux

---

There are no exercises in Chapter 1.

## Chapter 2: Creating the Perfect Linux Desktop

---

This section details some ways that these tasks can be completed on both the GNOME 2 and GNOME 3 desktops.

1. To get started, you need a Linux system in front of you to do the procedures in this book. An installed system is preferable, so you don't lose your changes when you reboot. To start out, you can use a Fedora Live CD (or installed system), an Ubuntu installed system, or a Red Hat Enterprise Linux installed system. Here are your choices:
  - a. **Fedora Live CD (GNOME 3):** Get a Fedora Live CD as described in Appendix A. Run it live, as described in the section "Starting with the Fedora GNOME Desktop Live image" in Chapter 2, or install it and run it from hard disk as described in Chapter 9, "Installing Linux."
  - b. **Ubuntu (GNOME 3):** Install Ubuntu and the GNOME Shell software, as described at the beginning of Chapter 2.
  - c. **Red Hat Enterprise Linux 8 (GNOME 3):** Install Red Hat Enterprise Linux 7, as described in Chapter 9.
  - d. **Red Hat Enterprise Linux 6 or earlier (GNOME 2):** Install Red Hat Enterprise Linux 6.

2. To launch the Firefox web browser and go to the GNOME home page (<http://gnome.org>), there are some easy steps to take. If your network is not working, refer to Chapter 14, “Administering Networking,” for help on connecting to wired and wireless networks.

### GNOME 3

For GNOME 3, you can press the Windows key to get to the Overview screen. Then type **Firefox** to highlight just the Firefox web browser icon. Press Enter to launch it. Type <http://gnome.org> in the location box, and press Enter.

### GNOME 2

For GNOME 2, select the Firefox icon from the top menu bar. Type <http://gnome.org> in the location box, and press Enter.

3. To pick a background that you like from the GNOME art site (<http://gnome-look.org>), download it to your Pictures folder, and select it as your current background. On both GNOME 2 and GNOME 3 systems, do the following:
  - a. Type <http://gnome-look.org/> in the Firefox location box and press Enter.
  - b. Find a background that you like and select it. Then click the Download button and download it to your Pictures folder.
  - c. Open your Pictures folder, right-click the image, and select Set as Wallpaper. The image is used as your desktop background.
4. To start a Nautilus File Manager window and move it to the second workspace on your desktop, do the following:

#### For GNOME 3

- a. Press the Windows key.
- b. Select the Files icon from the Dash (left side). A new instance of Nautilus starts in the current workspace.
- c. Right-click the title bar in the Files window and select Move to Monitor Down. The Files window moves to the second workspace.

#### For GNOME 2

- a. Open the Home folder from the GNOME 2 desktop (double-click).
- b. Right-click in the Nautilus title bar that appears, and select either Move to Workspace Right or Move to Another Workspace. (You can select which workspace you want from the list.)
5. To find the image that you downloaded to use as your desktop background and open it in any image viewer, first go to your Home folder, then open the Pictures folder. Double-click the image to open it in an image viewer.
6. Moving back and forth between the workspace with Firefox on it and the one with the Nautilus file manager is fairly straightforward.

If you did the previous exercises properly, Nautilus and Firefox should be in different workspaces. Here's how you can move between those workspaces in GNOME 3 and GNOME 2:

**GNOME 3**

Press the Windows key, and select the workspace that you want in the right column. As an alternative, you can go directly to the application that you want by pressing Alt+Tab and pressing Tab again and also arrow keys to highlight the application that you want to open.

**GNOME 2**

Select the workspace that you want with your mouse by clicking the small representation of the workspace in the right side of the lower panel. If you happen to have Desktop Effects enabled (System ⇄ Preferences Desktop Effects ⇄ Compiz), try pressing Ctrl+Alt+right arrow (or left arrow) to spin to the next workspace.

7. To open a list of applications installed on your system and select an image viewer to open from that list using as few clicks or keystrokes as possible, do the following:

**In GNOME 3**

Move the mouse to the upper-left corner of the screen to get to the Overview screen. Select Applications, then select Utilities from the right column, and then select Image Viewer.

**In GNOME 2**

Select Applications ⇄ Graphics ⇄ Image Viewer to open an image viewer window on the desktop.

8. To change the view of the windows on your current workspace to smaller views of those windows that you can step through, do the following:

**In GNOME 3**

With multiple windows open on multiple workspaces, press the Alt+Tab keys. While continuing to hold the Alt key, press Tab until you highlight the application that you want. Release the Alt key to select it.

**In GNOME 2**

With multiple windows open on multiple workspaces, press and hold the Ctrl+Alt+Tab keys. While continuing to hold the Ctrl+Alt keys, press Tab until you have highlighted the application that you want. Release the Ctrl and Alt keys to select it.

9. To launch a music player from your desktop using only the keyboard, do the following:

**In GNOME 3**

- a. Press the Windows key to go to the Overview screen.

- b. Type **Rhyth** (until the icon appears and is highlighted) and press Enter.  
(In Ubuntu, if you don't have Rhythmbox installed, type **Bansh** to open the Banshee Media Player.)

### In GNOME 2

Press Alt+F2. From the Run Application box that appears. Then type **rhythmbox** and press Enter.

10. To take a picture of your desktop using only keystrokes, press the Print Screen key to take a screen shot of your entire desktop in both GNOME 3 and GNOME 2. Press Alt+Print Screen to take a screen shot of just the current window. In both cases, the images are saved to the Pictures folder in your home folder.

## Chapter 3: Using the Shell

---

1. To switch virtual consoles and return to the desktop in Fedora or Ubuntu (this feature is disabled in some RHEL systems), do the following:
  - a. Hold Ctrl+Alt and press F2 (Ctrl+Alt+F2). A text-based console should appear.
  - b. Type your username (press Enter) and password (press Enter).
  - c. Type a few commands, such as **id**, **pwd**, and **ls**.
  - d. Type **exit** to exit the shell and return to the login prompt.
  - e. Press Ctrl+Alt+F1 to return to the virtual console that holds your desktop.  
(On different Linux systems, the desktop may be on different virtual consoles. Ctrl+Alt+F7 and Ctrl+Alt+F2 are other common places to find it.)
2. For your Terminal window, make the font red and the background yellow.
  - a. From the GNOME desktop, select Applications ⇨ System Tools ⇨ Terminal to open a Terminal window.
  - b. From the Terminal window, select Edit ⇨ Profile Preferences.
  - c. Select the Colors tab and deselect "Use colors from system theme" box.
  - d. Select the box next to Text Color, click the color red that you want from the available selections, and click Select.
  - e. Select the box next to Background Color, click the color yellow that you want from the available selections, and click Select.
  - f. Click Close on the Profile window to go back to the Terminal window with the new colors.
  - g. Go back and reselect "Use colors from system theme" box to go back to the default Terminal colors.
3. Find the `mount` command and `tracpath` man page.
  - a. Run `type mount` to see that the `mount` command's location is either `/usr/bin/mount` or `/bin/mount`.

- b. Run `locate tracepath` to see that the `tracepath` man page is at `/usr/share/man/man8/tracepath.8.gz`.

4. Run, recall, and change these commands as described:

```
$ cat /etc/passwd
$ ls $HOME
$ date
```

- a. Press the up arrow until you see the `cat /etc/passwd` command. If your cursor is not already at the end of the line, press `Ctrl+E` to get there. Backspace over the word `passwd`, type the word **group**, and press Enter.
- b. Type **man ls**, and find the option to list by time (`-t`). Press the up arrow until you see the `ls $HOME` command. Use the left arrow key or `Alt+B` to position your cursor to the left of `$HOME`. Type `-t`, so that the line appears as `ls -t $HOME`. Press Enter to run the command.
- c. Type **man date** to view the date man page. Use the up arrow to recall the date command and add the format indicator that you found. A single `%D` format indicator gets the results you need:

```
$ date +%D
04/27/20
```

5. Use tab completion to type `basename /usr/share/doc/`. Type `basen<Tab> /u<Tab>sh<Tab>do<Tab>` to get `basename/usr/share/doc/`.
6. Pipe `/etc/services` to the `less` command: `$ cat /etc/services | less`.
7. Make output from the `date` command appear in this format: Today is Thursday, April 23, 2020.

```
$ echo "Today is $(date +%A, %B %d, %Y)"
```

8. View variables to find your current hostname, username, shell, and home directories.

```
$ echo $HOSTNAME
$ echo $USERNAME
$ echo $SHELL
$ echo $HOME
```

9. Add a permanent `mypass` alias that displays the contents of the `/etc/passwd` file.

- a. Type `nano $HOME/.bashrc`.
- b. Move the cursor to an open line at the bottom of the page. (Press Enter to open a new line if needed.)
- c. On its own line, type `alias m="cat /etc/passwd"`.
- d. Type `Ctrl+O` to save and `Ctrl+X` to exit the file.
- e. Type `source $HOME/.bashrc`.

- f. Type **alias m** to make sure that the alias was set properly: `alias m='cat /etc/passwd'`.
  - g. Type **m**. (The `/etc/passwd` file displays on the screen.)
10. To display the man page for the mount system call, use the `man -k` command to find man pages that include the word `mount`. Then use the `mount` command with the correct section number (8) to get the proper mount man page:

```
$ man -k mount | grep ^mount
mount          (2)  - mount filesystem
mount          (8)  - mount a filesystem
...
mountpoint     (1)  - see if a directory is a mountpoint
mountstats     (8)  - Displays various NFS client per-mount
statistics
$ man 2 mount
MOUNT(2)       Linux Programmer's Manual
MOUNT(2)
NAME
    mount - mount file system
SYNOPSIS
    #include <sys/mount.h>
.
.
.
```

## Chapter 4: Moving Around the Filesystem

---

1. Create the `projects` directory, create nine empty files (`house1` to `house9`), and list just those files.

```
$ mkdir $HOME/projects/
$ touch $HOME/projects/house{1..9}
$ ls $HOME/projects/house{1..9}
```

2. Make the `$HOME/projects/houses/doors/` directory path, and create some empty files in that path.

```
$ cd
$ mkdir $HOME/projects/houses
$ touch $HOME/projects/houses/bungalow.txt
$ mkdir $HOME/projects/houses/doors/
$ touch $HOME/projects/houses/doors/bifold.txt
$ mkdir -p $HOME/projects/outdoors/vegetation/
$ touch $HOME/projects/outdoors/vegetation/landscape.txt
```

3. Copy the files `house1` and `house5` to the `$HOME/projects/houses/` directory.

```
$ cp $HOME/projects/house[15] $HOME/projects/houses
```

4. Recursively copy the `/usr/share/doc/initscripts*` directory to the `$HOME/projects/` directory.

```
$ cp -ra /usr/share/doc/initscripts* $HOME/projects/
```

5. Recursively list the contents of the `$HOME/projects/` directory. Pipe the output to the `less` command so that you can page through the output.

```
$ ls -lR $HOME/projects/ | less
```

6. Remove the files `house6`, `house7`, and `house8` without being prompted.

```
$ rm -f $HOME/projects/house[678]
```

7. Move `house3` and `house4` to the `$HOME/projects/houses/doors` directory.

```
$ mv $HOME/projects/house{3,4} $HOME/projects/houses/doors/
```

8. Remove the `$HOME/projects/houses/doors` directory and its contents.

```
$ rm -rf $HOME/projects/houses/doors/
```

9. Change the permissions on the `$HOME/projects/house2` file so that it can be read and written to by the user who owns the file, only read by the group, and have no permission for others.

```
$ chmod 640 $HOME/projects/house2
```

10. Recursively change the permissions of the `$HOME/projects/` directory so that nobody has write permission to any files or directories beneath that point in the file system.

```
$ chmod -R a-w $HOME/projects/
$ ls -lR $HOME/projects/
/home/joe/projects/:
```

```
total 12
```

```
-r--r--r--. 1 joe joe    0 Jan 16 06:49 house1
```

```
-r--r-----. 1 joe joe    0 Jan 16 06:49 house2
```

```
-r--r--r--. 1 joe joe    0 Jan 16 06:49 house5
```

```
-r--r--r--. 1 joe joe    0 Jan 16 06:49 house9
```

```
dr-xr-xr-x. 2 joe joe 4096 Jan 16 06:57 houses
```

```
dr-xr-xr-x. 2 joe joe 4096 Jul  1 2014 initscripts-9.03.40
```

```
dr-xr-xr-x. 3 joe joe 4096 Jan 16 06:53 outdoors
```

```
...
```



## Chapter 5: Working with Text Files

---

1. Follow these steps to create the `/tmp/services` file, and then edit it so that `WorldWideWeb` appears as `World Wide Web`.

```
$ cp /etc/services /tmp
$ vi /tmp/services
/WorldWideWeb<Enter>
cwWorld Wide Web<Esc>
```

The next two lines show the before and after:

```
http          80/tcp      www www-http    # WorldWideWeb HTTP
http          80/tcp      www www-http    # World Wide Web HTTP
```

2. One way to move the paragraph in your `/tmp/services` file is to search for the first line of the paragraph, delete five lines (5dd), go to the end of the file (G), and put in the text (p):

```
$ vi /tmp/services
/Note that it is<Enter>
5dd
G
p
```

3. To use `ex` mode to search for every occurrence of the term `tcp` (case sensitive) in your `/tmp/services` file, and change it to `WHATEVER`, you can enter the following:

```
$ vi /tmp/services
:g/tcp/s//WHATEVER/g<Enter>
```

4. To search the `/etc` directory for every file named `passwd` and redirect errors from your search to `/dev/null`, you can enter the following:

```
$ find /etc -name passwd 2> /dev/null
```

5. Create a directory in your home directory called `TEST`. Create files in that directory named `one`, `two`, and `three` that have full read/write/execute permissions on for everyone (user, group, and other). Construct a `find` command that would find those files and any other files that have write permission open to “others” from your home directory and below.

```
$ mkdir $HOME/TEST
$ touch $HOME/TEST/{one,two,three}
$ chmod 777 $HOME/TEST/{one,two,three}
$ find $HOME -perm -002 -type f -ls
148120 0 -rwxrwxrwx 1 chris chris 0 Jan  1 08:56 /home/
chris/TEST/two
```

```

148918 0 -rwxrwxrwx 1 chris chris 0 Jan 1 08:56 home/chris/
TEST/three
147306 0 -rwxrwxrwx 1 chris chris 0 Jan 1 08:56 /home/chris/
TEST/one

```

- Find files under the `/usr/share/doc` directory that have not been modified in more than 300 days.

```
$ find /usr/share/doc -mtime +300
```

- Create a `/tmp/FILES` directory. Find all files under the `/usr/share` directory that are more than 5MB and less than 10MB, and copy them to the `/tmp/FILES` directory.

```

$ mkdir /tmp/FILES
$ find /usr/share -size +5M -size -10M -exec cp {} /tmp/FILES \;
$ du -sh /tmp/FILES/*
6.6M    /tmp/FILES/BidiCharacterTest.txt
7.6M    /tmp/FILES/BidiTest.txt
5.2M    /tmp/FILES/day.jpg

```

- Find every file in the `/tmp/FILES` directory, and make a backup copy of each file in the same directory. Use each file's existing name and append `.mybackup` to create each backup file.

```
$ find /tmp/FILES/ -type f -exec cp {} {}.mybackup \;
```

- Install the `kernel-doc` package in Fedora or Red Hat Enterprise Linux. Using `grep`, search inside the files contained in the `/usr/share/doc/kernel-doc*` directory for the term `e1000` (case insensitive), and list the names of the files that contain that term.

```

# yum install kernel-doc
$ cd /usr/share/doc/kernel-doc*
$ grep -rli e1000 .
./Documentation/powerpc/booting-without-of.txt
./Documentation/networking/e100.txt
...

```

- Search for the `e1000` term again in the same location. However, this time list every line that contains the term and highlight the term in color.

```

$ cd /usr/share/doc/kernel-doc-*
$ grep -ri --color e1000 .

```

## Chapter 6: Managing Running Processes

- To list all processes running on your system with a full set of columns, while piping the output to `less`, enter the following:

```
$ ps -ef | less
```

2. To list all processes running on the system and sort those processes by the name of the user running each process, enter the following:

```
$ ps -ef --sort=user | less
```

3. To list all processes running on the system with the column names process ID, username, group name, nice value, virtual memory size, resident memory size, and command, enter the following:

```
$ ps -eo 'pid,user,group,nice,vsz,rss,comm' | less
PID USER      GROUP      NI    VSZ   RSS  COMMAND
   1 root        root        0  19324  1236  init
   2 root        root        0      0      0  kthreadd
   3 root        root        -      0      0  migration/0
   4 root        root        0      0      0  ksoftirqd/0
```

4. To run the `top` command and then go back and forth between sorting by CPU usage and memory consumption, enter the following:

```
$ top
P
M
P
M
```

5. To start the `gedit` process from your desktop and use the System Monitor window to kill that process, do the following:

```
$ gedit &
```

Next, in GNOME 2, select Applications ⇨ System Tools ⇨ System Monitor, or in GNOME 3, from the Activities screen, type **System Monitor** and press Enter. Find the `gedit` process on the Processes tab. (You can sort alphabetically to make it easier by clicking the Process Name heading.) Right-click the `gedit` command, and then select either End Process or Kill Process; the `gedit` window on your screen should disappear.

6. To run the `gedit` process and use the `kill` command to send a signal to pause (stop) that process, enter the following:

```
$ gedit &
[1] 21532

$ kill -SIGSTOP 21532
```

7. To use the `killall` command to tell the `gedit` command (paused in the previous exercise) to continue working, do the following:

```
$ killall -SIGCONT gedit
```

Make sure that the text you typed after `gedit` was paused now appears in the window.

8. To install the `xeyes` command, run it about 20 times in the background, and run `killall` to kill all 20 `xeyes` processes at once, enter the following:

```
# yum install xorg-x11-apps
$ xeyes &
$ xeyes &
...
$ killall xeyes &
```

Remember, you need to be the root user to install the package. After that, remember to repeat the `xeyes` command 20 times. Spread the windows around on your screen, and move the mouse for fun to watch the eyes move. All the `xeyes` windows should disappear at once when you type `killall xeyes`.

9. As a regular user, run the `gedit` command so that it starts with a nice value of 5.

```
# nice -n 5 gedit &
[1] 21578
```

10. To use the `renice` command to change the nice value of the `gedit` command you just started to 7, enter the following:

```
# renice -n 7 21578
21578: old priority 0, new priority 7
```

Use any command you like to verify that the current nice value for the `gedit` command is now set to 7. For example, you could type the following:

```
# ps -eo 'pid,user,nice,comm' | grep gedit
21578 chris      7 gedit
```

## Chapter 7: Writing Simple Shell Scripts

1. Here's an example of how to create a script in your `$HOME/bin` directory called `myownscript`. When the script runs, it should output information that appears as follows:

```
Today is Sat Jun 10 15:45:04 EDT 2019.
You are in /home/joe and your host is abc.example.com.
```

The following steps show one way to create the script named `myownscript`:

- a. If it doesn't already exist, create a `bin` directory:

```
$ mkdir $HOME/bin
```

- b. Using any text editor, create a script called `$HOME/bin/myownscript` that contains the following:

```
#!/bin/bash
# myownscript
# List some information about your current system
```

```
echo "Today is $(date)."
echo "You are in $(pwd) and your host is $(hostname)."
```

- c. Make the script executable:

```
$ chmod 755 $HOME/bin/myownscript
```

2. Create a script that reads in three positional parameters from the command line, assigns those parameters to variables named `ONE`, `TWO`, and `THREE`, respectively. Also, replace `X` with the number of parameters and `Y` with all of the parameters entered. Then replace `A` with the contents of variable `ONE`, `B` with variable `TWO`, and `C` with variable `THREE`, as shown below:

- a. To create the script, open a file named `$HOME/bin/myposition` and add the following contents:

```
#!/bin/bash
# myposition
ONE=$1
TWO=$2
THREE=$3
echo "There are $# parameters that include: $@"
echo "The first is $ONE, the second is $TWO, the third is $THREE."
```

- b. To make the script called `$HOME/bin/myposition` executable, enter the following:

```
$ chmod 755 $HOME/bin/myposition
```

- c. To test it, run it with some command-line arguments, as in the following:

```
$ myposition Where Is My Hat Buddy?
There are 5 parameters that include: Where Is My Hat Buddy?
The first is Where, the second is Is, the third is My.
```

3. To create the script described, do the following:

- a. To create a file called `$HOME/bin/myhome` and make it executable, enter the following:

```
$ touch $HOME/bin/myhome
$ chmod 755 $HOME/bin/myhome
```

- b. Here's what the script `myhome` might look like:

```
#!/bin/bash
# myhome
read -p "What street did you grow up on? " mystreet
read -p "What town did you grow up in? " mytown
echo "The street I grew up on was $mystreet and the town
was $mytown."
```

- c. Run the script to check that it works. The following example shows what the input and output for the script could look like:

```
$ myhome
What street did you grow up on? Harrison
What town did you grow up in? Princeton
The street I grew up on was Harrison and the town was Princeton.
```

4. To create the required script, do the following:

- a. Using any text editor, create a script called `$HOME/bin/myos` and make the script executable:

```
$ touch $HOME/bin/myos
$ chmod 755 $HOME/bin/myos
```

- b. The script could contain the following:

```
#!/bin/bash
# myos
read -p "What is your favorite operating system, Mac,
Windows or Linux? " opsys
if [ $opsys = Mac ] ; then
    echo "Mac is nice, but not tough enough for me."
elif [ $opsys = Windows ] ; then
    echo "I used Windows once. What is that blue screen
for?"
elif [ $opsys = Linux ] ; then
    echo "Great Choice!"
else
    echo "Is $opsys an operating system?"
fi
```

5. To create a script named `$HOME/bin/animals` that runs through the words *moose*, *cow*, *goose*, and *sow* through a `for` loop and have each of those words appended to the end of the line "I have a . . ." do the following:

- a. Make the script executable:

```
$ touch $HOME/bin/animals
$ chmod 755 $HOME/bin/animals
```

- b. The script could contain the following:

```
#!/bin/bash
# animals
for ANIMALS in moose cow goose sow ; do
    echo "I have a $ANIMALS"
done
```

- c. When you run the script, the output should appear as follows:

```
$ animals
I have a moose
```

```
I have a cow
I have a goose
I have a sow
```

## Chapter 8: Learning System Administration

---

1. To enable Cockpit on your system, enter the following:

```
# systemctl enable --now cockpit.socket
Created symlink /etc/systemd/system/sockets.target.wants/
cockpit.socket
→ /usr/lib/systemd/system/cockpit.socket
```

2. To open the Cockpit interface in your web browser, enter the hostname or IP address of the system holding your Cockpit service, followed by port number 9090. For example, enter this into the location box of your browser:

```
https://host1.example.com:9090/
```

3. To find all of the files under the `/var/spool` directory that are owned by users other than root and do a long listing of them, enter the following. (I recommend becoming root to find files that might be closed off to other users.)

```
$ su -
Password: *****
# find /var/spool -not -user root -ls | less
```

4. To become root user and create an empty or plain-text file named `/mnt/test.txt`, enter the following:

```
$ su -
Password: *****
# touch /mnt/test.txt
# ls -l /mnt/test.txt
-rw-r--r--. 1 root root 0 Jan  9 21:51 /mnt/test.txt
```

5. To become root and edit the `/etc/sudoers` file to allow your regular user account (for example, `bill`) to have full root privilege via the `sudo` command, do the following:

```
$ su -
Password: *****
# visudo
o
bill      ALL=(ALL)      ALL
Esc ZZ
```

Because `visudo` opens the `/etc/sudoers` file in `vi`, the example types `o` to open a line, and then it types in the line to allow `bill` to have full root privilege.

After the line is typed, press Esc to return to command mode and type **zz** to write and quit.

6. To use the `sudo` command to create a file called `/mnt/test2.txt` and verify that the file is there and owned by the root user, enter the following:

```
[bill]$ sudo touch /mnt/test2.txt
We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:
    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.
[sudo] password for bill: *****
[bill]$ ls -l /mnt/test2.txt
-rw-r--r--. 1 root root 0 Jan  9 23:37 /mnt/test2.txt
```

7. Do the following to mount and unmount a USB drive and watch the system journal during this process:

- a. Run the `journalctl -f` command as root in a Terminal window and watch the output from here for the next few steps.

```
# journalctl -f
Jan 25 16:07:59 host2 kernel: usb 1-1.1: new high-speed USB device
number 16 using ehci-pci
Jan 25 16:07:59 host2 kernel: usb 1-1.1: New USB device found,
idVendor=0ea0, idProduct=2168
Jan 25 16:07:59 host2 kernel: usb 1-1.1: New USB device strings:
Mfr=1, Product=2, SerialNumber=3
Jan 25 16:07:59 host2 kernel: usb 1-1.1: Product: Flash Disk
Jan 25 16:07:59 host2 kernel: usb 1-1.1: Manufacturer: USB
...
Jan 25 16:08:01 host2 kernel: sd 18:0:0:0: [sdb] Write Protect is off
Jan 25 16:08:01 host2 kernel: sd 18:0:0:0: [sdb]
Assuming drive cache: write through
Jan 25 16:08:01 host2 kernel: sdb: sdb1
Jan 25 16:08:01 host2 kernel: sd 18:0:0:0: [sdb]
Attached SCSI removable disk
```

- b. Plug in a USB storage drive that mounts a filesystem from that drive automatically. If it does not, run the following commands in a second terminal (as root) to create a mount point directory and mount the device:

```
$ mkdir /mnt/test
$ mount /dev/sdb1 /mnt/test
$ umount /dev/sdb1
```



8. To see what USB devices are connected to your computer, enter the following:

```
$ lsusb
```

9. To load the `bttv` module, list the modules that were loaded, and unload it, enter the following:

```
# modprobe -a bttv
# lsmod | grep bttv
ttv                167936  0
tea575x            16384  1 bttv
tveeprom           28672  1 bttv
videobuf_dma_sg    24576  1 bttv
videobuf_core      32768  2 videobuf_dma_sg,bttv
v4l2_common        16384  1 bttv
videodev           233472  3 tea575x,v4l2_common,bttv
i2c_algo_bit       16384  1 bttv
```

Notice that other modules (`v4l2_common`, `videodev`, and others) were loaded when you loaded `bttv` with `modprobe -a`.

10. Enter the following to remove the `bttv` module along with any other modules that were loaded with it. Notice that they were all gone after running `modprobe -r`.

```
# modprobe -r bttv
# lsmod | grep bttv
```

## Chapter 9: Installing Linux

---

1. To install a Fedora system from Fedora Live media, follow the instructions in the section “Installing Fedora from Live Media” in Chapter 9. In general, those steps include the following:
  - a. Booting the Live media.
  - b. Selecting to install to the hard drive when the system boots up.
  - c. Adding information from the summary page needed to configure your system initially.
  - d. Rebooting your computer and removing the Live medium so that the newly installed system boots from the hard drive.
2. To update the packages, after the Fedora Live media installation is complete, do the following:
  - a. Reboot the computer and fill in the first boot questions as prompted.
  - b. Using a wired or wireless connection, make sure that you have a connection to the Internet. Refer to Chapter 14, “Administering Networking,” if you have trouble getting your networking connection to work properly. Open a shell as the root user and type `sudo dnf update`.
  - c. When prompted, type `y` to accept the list of packages displayed. The system begins downloading and installing the packages.

3. To run the RHEL installation in text mode, do the following:
  - a. Boot the RHEL DVD.
  - b. When you see the boot menu, highlight one of the installation boot entries and press Tab. Move the cursor right to the end of the kernel line, and type the literal option `text` at the end of that line. Press Enter to start the installer.
  - c. Try out the rest of the installation in text mode.
4. To set the disk partitioning as described in question 4 for a Red Hat Enterprise Linux DVD installation, do the following:

**NOTE**

This procedure ultimately deletes all content on your hard disk. If you just want to use this exercise to practice partitioning, you can reboot your computer before starting the actual installation process without harming your hard disk. After you go forward and partition your disk, assume that all data has been deleted.

- a. On a computer that you can erase with at least 10GB of disk space, insert a RHEL installation DVD, reboot, and begin stepping through the installation screens.
- b. When you get to the Installation Summary screen, select Installation Destination.
- c. From the Installation Destination screen, select the device to use for the installation (probably `sda` if you have a single hard disk that you can completely erase or `vda` for a virtual install).
- d. Select the Custom button.
- e. Select Done to get to the Manual Partitioning screen.
- f. If the existing disk space is already consumed, you need to delete the partitions before proceeding.
- g. Click the plus (+) button at the bottom of the screen. Then add each of the following mount points:

```
/boot - 400M
/ - 3G
/var - 2G
/home -2G
```
- h. Select Done. You should see a summary of changes.
- i. If the changes look acceptable, select Accept Changes. If you are just practicing and don't actually want to change your partitions, select Cancel & Return to Custom Partitioning. Then simply exit the installer.

## Chapter 10: Getting and Managing Software

---

1. To search the YUM repository for the package that provides the `mogrify` command, enter the following:

```
# yum provides mogrify
```

2. To display information about the package that provides the `mogrify` command and determine what is that package's home page (URL), enter the following:

```
# yum info ImageMagick
```

You will see that the URL to the home page for ImageMagick is <http://www.imagemagick.org>.

3. To install the package containing the `mogrify` command, enter the following:

```
# yum install ImageMagick
```

4. To list all of the documentation files contained in the package that provides the `mogrify` command, enter the following:

```
# rpm -qd ImageMagick
...
/usr/share/doc/ImageMagick/README.txt
...
/usr/share/man/man1/identify.1.gz
/usr/share/man/man1/import.1.gz
/usr/share/man/man1/mogrify.1.gz
```

5. To look through the change log of the package that provides the `mogrify` command, enter the following:

```
# rpm -q --changelog ImageMagick | less
```

6. To delete the `mogrify` command from your system and verify its package against the RPM database to see that the command is indeed missing, enter the following:

```
# type mogrify
mogrify is /usr/bin/mogrify
# rm /usr/bin/mogrify
rm remove regular file '/usr/bin/mogrify'? y
# rpm -V ImageMagick
missing    /usr/bin/mogrify
```

7. To reinstall the package that provides the `mogrify` command and make sure that the entire package is intact again, enter the following:

```
# yum reinstall ImageMagick
# rpm -V ImageMagick
```

8. To download the package that provides the `mogrify` command to your current directory, enter the following:

```
# yum download ImageMagick
ImageMagick-6.9.10.28-1.fc30.x86_64.rpm
```

9. To display general information about the package that you just downloaded by querying the package's RPM file in the current directory, enter the following:

```
# rpm -qip ImageMagick-6.9.10.28-1.fc30.x86_64.rpm
Name       : ImageMagick
Epoch     : 1
Version    : 6.9.10.28
Release    : 1.fc30
```

...

10. To remove the package containing the `mogrify` command from your system, enter the following:

```
# yum remove ImageMagick
```

## Chapter 11: Managing User Accounts

For questions that involve adding and removing user accounts, you can use the Users window, the User Manager window, or command-line tools such as `useradd` and `usermod`. The point is to make sure that you get the correct results shown in the answers that follow, not necessarily to do it exactly in the same way that I did.

There are multiple ways that you can achieve the same results. The answers here show how to complete the exercises from the command line. (Become root user when you see a `#` prompt.)

1. To add a local user account to your Linux system that has a username of `jbaxter` and a full name of John Baxter, which uses `/bin/sh` as its default shell and is the next available UID (yours may differ from the one shown here), enter the following. You can use the `grep` command to check the new user account. Then set the password for `jbaxter` to: `My1N1te0ut!`

```
# useradd -c "John Baxter" -s /bin/sh jbaxter
# grep jbaxter /etc/passwd
jbaxter:x:1001:1001:John Baxter:/home/jbaxter:/bin/sh
# passwd jbaxter
Changing password for user jbaxter
New password: My1N1te0ut!
Retype new password: My1N1te0ut!
passwd: all authentication tokens updated successfully
```

2. To create a group account named `testing` that uses group ID 315, enter the following:

```
# groupadd -g 315 testing
# grep testing /etc/group
testing:x:315:
```

3. To add jbxaxter to the testing group and the bin group, enter the following:

```
# usermod -aG testing,bin jbxaxter
# grep jbxaxter /etc/group
bin:x:1:bin,daemon,jbxaxter
jbxaxter:x:1001:
testing:x:315:jbxaxter
```

4. To become jbxaxter and temporarily have the testing group be jbxaxter's default group, run `touch /home/jbxaxter/file.txt` so that the testing group is assigned as the file's group, and do the following:

```
$ su - jbxaxter
Password: My1N1te0ut!
sh-4.2$ newgrp testing
sh-4.2$ touch /home/jbxaxter/file.txt
sh-4.2$ ls -l /home/baxter/file.txt
-rw-rw-r--. 1 jbxaxter testing 0 Jan 25 06:42 /home/jbxaxter/file.
txt
sh-4.2$ exit ; exit
```

5. Note what user ID has been assigned to jbxaxter, and then delete the user account without deleting the home directory assigned to jbxaxter.

```
$ userdel jbxaxter
```

6. Use the following command to find any files in the `/home` directory (and any subdirectories) that are assigned to the user ID that recently belonged to the user named jbxaxter. (When I did it, the UID/GID were both 1001; yours may differ.) Notice that the username jbxaxter is no longer assigned on the system, so any files that user created are listed as belonging to UID 1001 and GID 1001, except for a couple of files that were assigned to the testing group because of the `newgrp` command run earlier:

```
# find /home -uid 1001 -ls
262184  4 drwx----- 4 1001  1001  4096 Jan 25 08:00 /home/
jbxaxter
262193  4 -rw-r--r--  1 1001  1001   176 Jan 27  2011 /home/
jbxaxter/.bash_profile
262196  4 -rw-----  1 13602 testing 93 Jan 25 08:00 /home/
jbxaxter/.bash_history
262194  0 -rw-rw-r--  1 13602 testing  0 Jan 25 07:59 /home/
jbxaxter/file.txt
...
```

7. Run these commands to copy the `/etc/services` file to the `/etc/skel/` directory; then add a new user to the system named mjones, with a full name of Mary Jones and a home directory of `/home/maryjones`. List her home directory to make sure that the services file is there.

```
# cp /etc/services /etc/skel/
# useradd -d /home/maryjones -c "Mary Jones" mjones
```

```
# ls -l /home/maryjones
total 628
-rw-r--r--. 1 mjones mjones 640999 Jan 25 06:27 services
```

8. Run the following command to find all files under the /home directory that belong to mjones. If you did the exercises in order, notice that after you deleted the user with the highest user ID and group ID, those numbers were assigned to mjones. As a result, any files left on the system by jbxaxter now belong to mjones. (For this reason, you should remove or change ownership of files left behind when you delete a user.)

```
# find /home -user mjones -ls
262184 4 drwx----- 4 mjones mjones 4096 Jan 25 08:00 /
home/jbxaxter
262193 4 -rw-r--r-- 1 mjones mjones 176 Jan 27 2011 /home/
jbxaxter/.bash_profile
262189 4 -rw-r--r-- 1 mjones mjones 18 Jan 27 2011 /home/
jbxaxter/.bash_logout
262194 0 -rw-rw-r-- 1 mjones testing 0 Jan 25 07:59 /home/
jbxaxter/file.txt
262188 4 -rw-r--r-- 1 mjones mjones 124 Jan 27 2011 /home/
jbxaxter/.bashrc
262197 4 drwx----- 4 mjones mjones 4096 Jan 25 08:27 /
home/maryjones
262207 4 -rw-r--r-- 1 mjones mjones 176 Jan 27 2011 /home/
maryjones/.bash_profile
262202 4 -rw-r--r-- 1 mjones mjones 18 Jan 27 2011 /home/
maryjones/.bash_logout
262206 628 -rw-r--r-- 1 mjones mjones 640999 Jan 25 08:27 /
home/maryjones/services
262201 4 -rw-r--r-- 1 mjones mjones 124 Jan 27 2011 /home/
maryjones/.bashrc
```

9. As the user mjones, you can use the following to create a file called /tmp/maryfile.txt, and use ACLs to assign the bin user read/write permission and the lp group read/write permission to that file.

```
[mjones]$ touch /tmp/maryfile.txt
[mjones]$ setfacl -m u:bin:rw /tmp/maryfile.txt
[mjones]$ setfacl -m g:lp:rw /tmp/maryfile.txt
[mjones]$ getfacl /tmp/maryfile.txt
# file: tmp/maryfile.txt
# owner: mjones
# group: mjones
user::rw-
user:bin:rw-
group::rw-
group:lp:rw-
mask::rw-
other::r& -
```

10. Run this set of commands (as mjones) to create a directory named `/tmp/mydir`, and use ACLs to assign default permissions to it so that the `adm` user has read/write/execute permission to that directory and any files or directories created in it. Test that it worked by creating the `/tmp/mydir/testing/` directory and `/tmp/mydir/newfile.txt`.

```
[mary]$ mkdir /tmp/mydir
[mary]$ setfacl -m d:u:adm:rwX /tmp/mydir
[mjones]$ getfacl /tmp/mydir
# file: tmp/mydir
# owner: mjones
# group: mjones
user::rwX
group::rwX
other::r-X
default:user::rwX
default:user:adm:rwX
default:group::rwX
default:mask::rwX
default:other::r-X
[mjones]$ mkdir /tmp/mydir/testing
[mjones]$ touch /tmp/mydir/newfile.txt
[mjones]$ getfacl /tmp/mydir/testing/
# file: tmp/mydir/testing/
# owner: mjones
# group: mjones
user::rwX
user:adm:rwX
group::rwX
mask::rwX
other::r-X
default:user::rwX
default:user:adm:rwX
default:group::rwX
default:mask::rwX
default:other::r-X
[mjones]$ getfacl /tmp/mydir/newfile.txt
# file: tmp/mydir/newfile.txt
# owner: mjones
# group: mjones
user::rw-
user:adm:rwX      #effective:rw-
group::rwX        #effective:rw-
mask::rw-
other::r--
```

Notice that the `adm` user effectively has only `rw-` permission. To remedy that, you need to expand the permissions of the mask. One way to do that is with the `chmod` command, as follows:

```
[mjones]$ chmod 775 /tmp/mydir/newfile.txt
[mjones]$ getfacl /tmp/mydir/newfile.txt
# file: tmp/mydir/newfile.txt
# owner: mjones
# group: mjones
user::rwx
user:adm:rwx
group::rwx
mask::rwx
other::r-x
```

## Chapter 12: Managing Disks and Filesystems

1. To determine the device name of a USB flash drive that you want to insert into your computer, enter the following and insert the USB flash drive. (Press `Ctrl+C` after you have seen the appropriate messages.)

```
# journalctl -f
kernel: [sdb] 15667200 512-byte logical blocks:
      (8.02 GB/7.47 GiB)
Feb 11 21:55:59 cnegus kernel: sd 7:0:0:0:
      [sdb] Write Protect is off
Feb 11 21:55:59 cnegus kernel: [sdb] Assuming
      drive cache: write through
Feb 11 21:55:59 cnegus kernel: [sdb] Assuming
      drive cache: write through
```

2. To list partitions on the USB flash drive on a RHEL 6 system, enter the following:

```
# fdisk -c -u -l /dev/sdb
```

To list partitions on a RHEL 7, RHEL 8, or Fedora system, enter the following:

```
# fdisk -l /dev/sdb
```

3. To delete partitions on the USB flash drive, assuming device `/dev/sdb`, do the following:

```
# fdisk /dev/sdb
Command (m for help): d
Partition number (1-6): 6
Command (m for help): d
Partition number (1-5): 5
Command (m for help): d
```



```
Partition number (1-5): 4
Command (m for help): d
Partition number (1-4): 3
Command (m for help): d
Partition number (1-4): 2
Command (m for help): d
Selected partition 1
Command (m for help): w
# partprobe /dev/sdb
```

4. To add a 100MB Linux partition, 200MB swap partition, and 500MB LVM partition to the USB flash drive, enter the following:

```
# fdisk /dev/sdb
```

```
Command (m for help): n
Command action
  e   extended
  p   primary partition (1-4)
p
Partition number (1-4): 1
First sector (2048-15667199, default 2048): <ENTER>
Last sector, +sectors or +size{K,M,G} (default 15667199): +100M
Command (m for help): n
Command action
  e   extended
  p   primary partition (1-4)
p
Partition number (1-4): 2
First sector (616448-8342527, default 616448): <ENTER>
Last sector, +sectors or +size{K,M,G} (default 15667199): +200M
Command (m for help): n
Command action
  e   extended
  p   primary partition (1-4)
p
Partition number (1-4): 3
First sector (616448-15667199, default 616448): <ENTER>
Using default value 616448
Last sector, +sectors or +size{K,M,G} (default 15667199): +500M
Command (m for help): t
Partition number (1-4): 2
Hex code (type L to list codes): 82
Changed system type of partition 2 to 82 (Linux swap / Solaris)
Command (m for help): t
Partition number (1-4): 3
Hex code (type L to list codes): 8e
Changed system type of partition 3 to 8e (Linux LVM)
```

```

Command (m for help): w
# partprobe /dev/sdb
# grep sdb /proc/partitions
 8      16      7833600 sdb
 8      17      102400 sdb1
 8      18      204800 sdb2
 8      19      512000 sdb3

```

5. To put an ext4 filesystem on the Linux partition, enter the following:

```
# mkfs -t ext4 /dev/sdb1
```

6. To create a mount point called /mnt/mypart and mount the Linux partition on it, do the following:

```
# mkdir /mnt/mypart
# mount -t ext4 /dev/sdb1 /mnt/mypart
```

7. To enable the swap partition and turn it on so that additional swap space is immediately available, enter the following:

```
# mkswap /dev/sdb2
# swapon /dev/sdb2
```

8. To create a volume group called abc from the LVM partition, create a 200MB logical volume from that group called data, create a VFAT filesystem on it, temporarily mount the logical volume on a new directory named /mnt/test, and then check that it was successfully mounted, enter the following:

```
# pvcreate /dev/sdb3
# vgcreate abc /dev/sdb3
# lvcreate -n data -L 200M abc
# mkfs -t vfat /dev/mapper/abc-data
# mkdir /mnt/test
# mount /dev/mapper/abc-data /mnt/test
```

9. To grow the logical volume from 200MB to 300MB, enter the following:

```
# lvextend -L +100M /dev/mapper/abc-data
# resize2fs -p /dev/mapper/abc-data
```

10. To remove the USB flash drive safely from the computer, do the following:

```
# umount /dev/sdb1
# swapoff /dev/sdb2
# umount /mnt/test
# lvremove /dev/mapper/abc-data
# vgremove abc
# pvremove /dev/sdb3
```

You can now safely remove the USB flash drive from the computer.

## Chapter 13: Understanding Server Administration

---

1. To log in to any account on another computer using the `ssh` command, enter the following and then enter the password when prompted:

```
$ ssh joe@localhost
joe@localhost's password:
*****
[joe]$
```

2. To display the contents of a remote `/etc/system-release` file and have its contents displayed on the local system using remote execution with the `ssh` command, do the following:

```
$ ssh joe@localhost "cat /etc/system-release"
joe@localhost's password: *****
Fedora release 30 (Thirty)
```

3. To use X11 forwarding to display a `gedit` window on your local system and then save a file on the remote home directory, do the following:

```
$ ssh -X joe@localhost "gedit newfile"
joe@localhost's password: *****
$ ssh joe@localhost "cat newfile"
joe@localhost's password: *****
This is text from the file I saved in joe's remote home
directory
```

4. To copy all of the files from the `/usr/share/selinux` directory recursively on a remote system to the `/tmp` directory on your local system in such a way that all of the modification times on the files are updated to the time on the local system when they are copied, do the following:

```
$ scp -r joe@localhost:/usr/share/selinux /tmp
joe@localhost's password:
*****
irc.pp.bz2                100% 9673      9.5KB/s   00:00
dcc.pp.bz2                100%  15KB   15.2KB/s   00:01
$ ls -l /tmp/selinux | head
total 20
drwxr-xr-x. 3 root root  4096 Apr 18 05:52 devel
drwxr-xr-x. 2 root root  4096 Apr 18 05:52 packages
drwxr-xr-x. 2 root root 12288 Apr 18 05:52 targeted
```

5. To copy all of the files from the `/usr/share/logwatch` directory recursively on a remote system to the `/tmp` directory on your local system in such a way that all of the modification times on the files from the remote system are maintained on the local system, try the following:

```
$ rsync -av joe@localhost:/usr/share/logwatch /tmp
joe@localhost's password: *****
```

```
receiving incremental file list
logwatch/
logwatch/default.conf/
logwatch/default.conf/logwatch.conf
$ ls -l /tmp/logwatch | head
total 16
drwxr-xr-x. 5 root root 4096 Apr 19 2011 default.conf
drwxr-xr-x. 4 root root 4096 Feb 28 2011 dist.conf
drwxr-xr-x. 2 root root 4096 Apr 19 2011 lib
```

6. To create a public/private key pair to use for SSH communications (no passphrase on the key), copy the public key file to a remote user's account with `ssh-copy-id`, and use key-based authentication to log in to that user account without having to enter a password, use the following code:

```
$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/joe/.ssh/id_
rsa): ENTER
/home/joe/.ssh/id_rsa already exists.
Enter passphrase (empty for no passphrase): ENTER
Enter same passphrase again: ENTER
Your identification has been saved in /home/joe/.ssh/id_
rsa.
Your public key has been saved in /home/joe/.ssh/id_rsa.
pub.
The key fingerprint is:
58:ab:c1:95:b6:10:7a:aa:7c:c5:ab:bd:f3:4f:89:1e joe@cnegus.
csb
The key's randomart image is:
...
$ ssh-copy-id -i ~/.ssh/id_rsa.pub joe@localhost
joe@localhost's password: *****
Now try logging into the machine, with "ssh 'joe@
localhost'",
and check in:
.ssh/authorized_keys
to make sure we haven't added extra keys that you weren't
expecting.
$ ssh joe@localhost
$ cat .ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAyN2Psp5/
LRUC9E8BDCx53yPUa0qoOPd

v6H4sF3vmn04V6E7D1iXpzwPzdo4rpvmR1ZiinHR2xGAEr2uZag7feKgLnww2KpcQ6S
iR7lZrOhQjV+SGb/a1dxrIeZqKMq1Tk07G4EvboIrp//9J47vI4l7iNu0x
RmjI3TTxa
DdCTbpG6J3uSJm1BKzdUtwb413x35W2bRgMI75aIdeBsDgQBBiOdu+zuTM
rXJj2viCA
```

```
XeJ7gIWrvBaMQdOSvSdlkX353tmIjmJheWdgCccM/1jKdoELpaevg9anCe/
yUP3so31
tTo4I+qTfzAQD5+66oqW0LgMkWVvfZI7dUz3WUPmcMw== chris@abc.
example.com
```

7. To create an entry in `/etc/rsyslog.conf` that stores all authentication messages at the info level and higher into a file named `/var/log/myauth`, do the following. Watch from one terminal as the data comes in.

```
# vim /etc/rsyslog.conf
authpriv.info                                /var/log/myauth
# service rsyslog restart
or
# systemctl restart rsyslog.service
<Terminal 1>                                <Terminal 2>
# tail -f /var/log/myauth                    $ ssh joe@
localhost
Apr 18 06:19:34 abc unix_chkpwd[30631]      joe@localhost's
password:
Apr 18 06:19:34 abc sshd[30631]             Permission
denied,try again
:pam_unix(sshd:auth):
authentication failure;logname= uid=501
euid=501 tty=ssh ruser= rhost=localhost
user=joe
Apr 18 06:19:34 abc sshd[30631]:
Failed password for joe from
127.0.0.1 port 5564 ssh2
```

8. To determine the largest directory structures under `/usr/share`, sort them from largest to smallest, and list the top 10 of those directories in terms of size using the `du` command, enter the following:

```
$ du -s /usr/share/* | sort -rn | head
527800 /usr/share/locale
277108 /usr/share/fonts
196232 /usr/share/help

134984 /usr/share/backgrounds
...
```

9. To show the space that is used and available from all of the filesystems currently attached to the local system, but exclude any `tmpfs` or `devtmpfs` filesystems by using the `df` command, enter the following:

```
$ df -h -x tmpfs -x devtmpfs
Filesystem      Size  Used Avail Use% Mounted on
/deev/sda4      20G   4.2G  16G   22% /
```

10. To find any files in the `/usr` directory that are more than 10MB in size, do the following:

```
$ find /usr -size +10M
/usr/lib/locale/locale-archive
/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.212.b04-0.fc30.
x86_64/jre/lib/rt.jar
/usr/libexec/cni/dhcp
/usr/libexec/gdb
/usr/libexec/gcc/x86_64-redhat-linux/9/ltol
/usr/libexec/gcc/x86_64-redhat-linux/9/cc1
```

## Chapter 14: Administering Networking

1. To use the desktop to check that NetworkManager has successfully started your network interface (wired or wireless), do the following:
  - a. Left-click the upper-right corner of your GNOME desktop to see the drop-down menu. Any active wired or wireless network connections should appear on that menu.
  - b. If it has not connected to the network, select from the list of wired or wireless networks available, and then enter the username and password, if prompted, to start an active connection.
2. To run a command to check the active network interfaces available on your computer, enter the following:
 

```
$ ifconfig
```

or

```
$ ip addr show
```
3. Try to contact `google.com` from the command line in a way that ensures that DNS is working properly:
 

```
$ ping google.com
Ctrl-C
```
4. To run a command to check the routes being used to communicate outside of your local network, enter the following:
 

```
$ route
```
5. To trace the route being taken to connect to `google.com`, use the `traceroute` command:
 

```
$ traceroute google.com
```
6. To view the network interfaces and related network activities for your Linux system through Cockpit, open a web browser to port 9090 using an IP address or hostname. For example: `https://localhost:9090/network`.

7. To create a host entry that allows you to communicate with your local host system using the name `myownhost`, edit the `/etc/hosts` file (`vi /etc/hosts`), and add `myownhost` to the end of the `localhost` entry so that it appears as follows (then `ping myownhost` to see if it worked):

```
127.0.0.1          localhost.localdomain localhost myownhost
# ping myownhost
Ctrl+C
```

8. To see the DNS name servers being used to resolve hostnames and IP addresses on your system (yours will be different than those shown below), enter the following:

```
# cat /etc/resolv.conf
nameserver 10.83.14.9
nameserver 10.18.2.10
nameserver 192.168.1.254
# dig google.com
...
google.com.      91941   IN      NS      ns3.google.com.
;; Query time: 0 msec
;; SERVER: 10.18.2.9#53(10.18.2.9)
;; WHEN: Sat Nov 23 20:18:56 EST 2019
;; MSG SIZE rcvd: 276
```

9. To create a custom route that directs traffic destined for the `192.168.99.0/255.255.255.0` network to some IP address on your local network, such as `192.168.0.5` (first ensuring that the `192.168.99` network is not being used at your location), do the following:

- a. Determine the name of your network interface, for example `enp4s0`. In that case, as root run the following commands:

```
# cd /etc/sysconfig/network-scripts
# vi route-enp4s0
```

- b. Add the following lines to that file:

```
ADDRESS0=192.168.99.0
NETMASK0=255.255.255.0
GATEWAY0=192.168.0.5
```

- c. Restart networking and run `route` to see that the route is active:

```
# systemctl restart NetworkManager
# route -n
Kernel IP routing table

```

Ref	Use	Destination	Gateway	Genmask	Flags	Metric
		192.168.0.1	0.0.0.0	255.255.255.0	U	600
0	0	enp4s0				

```

        192.168.99.0 192.168.0.5 255.255.255.0 UG 600
0      0 enp4s0

```

10. To check to see if your system has been configured to allow IPv4 packets to be routed between network interfaces on your system, enter the following:

```

# cat /proc/sys/net/ipv4/ip_forward
0

```

A 0 shows that IPv4 packet forwarding is disabled; a 1 shows that it is enabled.

## Chapter 15: Starting and Stopping Services

1. To determine which initialization daemon your server is currently using, consider the following:

- a. In most cases today, PID 1 appears as the `systemd` daemon:

```

# ps -ef | head
UID          PID  PPID  C STIME TTY          TIME CMD
root          1      0  0 17:01 ?           00:00:04 /usr/
lib/systemd/systemd --
                switched-root --system --deserialize 18

```

If you type `ps -ef` and PID 1 is `init`, it still might be the `systemd` daemon. Use the `strings` command to see if `systemd` is in use:

```

# strings /sbin/init | grep -i systemd
systemd.unit=
systemd.log_target=
systemd.log_level=
...

```

- b. Most likely, you have the Upstart, SysVinit, or BSD `init` daemon if your `init` daemon is not `systemd`. But double-check at <http://wikipedia.org/wiki/Init>.
2. The tools you use to manage services depend primarily on which initialization system is in use. Try to run the `systemctl` and `service` commands to determine the type of initialization script in use for the `ssh` service on your system:

- a. For `systemd`, a positive result, shown here, means that the `sshd` has been converted to `systemd`:

```

# systemctl status sshd.service
sshd.service - OpenSSH server daemon
    Loaded: loaded (/lib/systemd/system/sshd.service;
enabled)
    Active: active (running) since Mon, 20 Apr 2020
12:35:20...

```

- b. If you don't see positive results for the preceding test, try the following command for the SysVinit `init` daemon. A positive result here, along with



negative results for the preceding tests, means that `sshd` is still using the SysVinit daemon.

```
# service ssh status
sshd (pid 2390) is running...
```

3. To determine your server's previous and current runlevel, use the `runlevel` command. It still works on all init daemons:

```
$ runlevel
N 3
```

4. To change the default runlevel or target unit on your Linux server, you can do one of the following (depending upon your server's init daemon):

- a. For SysVinit, edit the file `/etc/inittab` and change the `#` in the line `id:#:initdefault:` to 2, 3, 4, or 5.
- b. For `systemd`, change the `default.target` to the desired `runlevel#.target`, where `#` is 2, 3, 4, or 5. The following shows you how to change the target unit to `runlevel3.target`.

```
# systemctl set-default runlevel3.target
Removed /etc/systemd/system/default.target.
Created symlink /etc/systemd/system/default.target →
/usr/lib/systemd/system/multi-user.target.
```

5. To list out services running (or active) on your server, you need to use different commands, depending upon the initialization daemon you are using.

- a. For SysVinit, use the `service` command as shown in this example:

```
# service --status-all | grep running | sort
anacron (pid 2162) is running...
atd (pid 2172) is running...
```

- b. For `systemd`, use the `systemctl` command, as follows:

```
# systemctl list-unit-files --type=service | grep -v disabled
UNIT FILE                                STATE
abrt-ccpp.service                       enabled
abrt-oops.service                       enabled
...
```

6. To list out the running (or active) services on your Linux server, use the appropriate command(s) determined in answer 5 for the initialization daemon that your server is using.

7. For each initialization daemon, the following command(s) show a particular service's current status:

- a. For SysVinit, the `service service_name status` command is used.
- b. For `systemd`, the `systemctl status service_name` command is used.

8. To show the status of the cups daemon on your Linux server, use the following:

- a. For the SysVinit:

```
# service cups status
cupsd (pid 8236) is running...
```

- b. For systemd:

```
# systemctl status cups.service
cups.service - CUPS Printing Service
Loaded: loaded (/lib/systemd/system/cups.service; enabled)
Active: active (running) since Tue, 05 May 2020 04:43:5...
Main PID: 17003 (cupsd)
CGroup: name=systemd:/system/cups.service
17003 /usr/sbin/cupsd -f
```

9. To attempt to restart the cups daemon on your Linux server, use the following:

- a. For SysVinit:

```
# service cups restart
Stopping cups: [ OK ]
```

- b. For systemd:

```
# systemctl restart cups.service
```

10. To attempt to reload the cups daemon on your Linux server, use the following:

- a. For SysVinit:

```
# service cups reload
Reloading cups: [ OK ]
```

- b. For systemd, this is a trick question. You cannot reload the cups daemon on a systemd Linux server!

```
# systemctl reload cups.service
Failed to issue method call: Job type reload is
not applicable for unit cups.service.
```

## Chapter 16: Configuring a Print Server

For questions that involve working with printers, you can use either graphical or command-line tools in most cases. The point is to make sure that you get the correct results, shown in the answers that follow. The answers here include a mix of graphical and command-line ways of solving the exercises. (Become root user when you see a # prompt.)

1. To use the Print Settings window to add a new printer called myprinter to your system (generic PostScript printer, connected to a port), do the following from Fedora 30:

- a. Install the system-config-printer package:

```
# dnf install system-config-printer
```

- b. From the GNOME 3 desktop, select Print Settings from the Activities screen.
  - c. Unlock the interface and enter the root password.
  - d. Select the Add button.
  - e. Select a USB or other port as the device and click Forward.
  - f. For the driver, choose Generic and click Forward; then choose PostScript and click Forward.
  - g. Click Forward to skip any installable options, if needed.
  - h. For the printer name, call it `myprinter`, give it any description and location you like, and click Apply.
  - i. Click Cancel in order not to print a test page. The printer should appear in the Print Settings window.
2. To use the `lpstat -t` command to see the status of all of your printers, enter the following:

```
# lpstat -t
deskjet-5550 accepting requests since Mon 02 Mar 2020
07:30:03 PM EST
```
3. To use the `lpr` command to print the `/etc/hosts` file, enter the following:

```
$ lp /etc/hosts -P myprinter
```
4. To check the print queue for that printer, enter the following:

```
# lpq -P myprinter
myprinter is not ready
Rank    Owner   Job      File(s)                Total Size
1st     root    655      hosts                  1024 bytes
```
5. To remove the print job from the queue (cancel it), enter the following.

```
# lprm -P myprinter
```
6. To use the printing window to set the basic server setting that publishes your printers so that other systems on your local network can print to your printers, do the following:
  - a. On a GNOME 3 desktop, from the Activities screen, type **Print Settings** and press Enter.
  - b. Select Server ⇄ Settings and type the root password if prompted.
  - c. Click the check box next to “Publish shared printers connected to this system” and click OK.

7. To allow remote administration of your system from a web browser, follow these steps:
  - a. On a GNOME 3 desktop, from the Activities screen, type **Print Settings**, and press Enter.
  - b. Select Server ⇨ Settings and type the root password if prompted.
  - c. Click the check box next to “Allow remote administration” and click OK.
8. To demonstrate that you can do remote administration of your system from a web browser on another system, do the following:
  - a. In the location box from a browser window from another computer on your network, enter the following, replacing *hostname* with the name or IP address of the system running your print service: **http://hostname:631**.
  - b. Type **root** as the user and the root password, when prompted. The CUPS home page should appear from that system.
9. To use the `netstat` command to see on which addresses the `cupsd` daemon is listening, enter the following:
 

```
# netstat -tupln | grep 631
tcp      0      0 0.0.0.0:631      0.0.0.0:*        LISTEN    6492/cupsd
tcp6     0      0 :::631          :::*             LISTEN    6492/cupsd
```
10. To delete the `myprinter` printer entry from your system, do the following:
  - a. Click the Unlock button and type the root password when prompted.
  - b. From the Print Settings window, right-click the `myprinter` icon and select Delete.
  - c. When prompted, select Delete again.

## Chapter 17: Configuring a Web Server

1. To install all of the packages associated with the Web Server group on a Fedora system, do the following:
 

```
# yum groupinstall "Web Server"
```
2. To create a file called `index.html` in the directory assigned to `DocumentRoot` in the main Apache configuration file (with the words “My Own Web Server” inside), do the following:
  - a. Determine the location of `DocumentRoot`:
 

```
# grep ^DocumentRoot /etc/httpd/conf/httpd.conf

DocumentRoot "/var/www/html"
```

- b. Echo the words “My Own Web Server” into the `index.html` file located in `DocumentRoot`:

```
# echo "My Own Web Server" > /var/www/html/index.html
```

3. To start the Apache web server and set it to start up automatically at boot time, then check that it is available from a web browser on your local host, do the following. (You should see the words “My Own Web Server” displayed if it is working properly.)

The `httpd` service is started and enabled differently on different Linux systems. In recent Fedora 30 or RHEL 7 or 8, enter the following:

```
# systemctl start httpd.service
# systemctl enable httpd.service
```

In RHEL 6 or earlier, enter the following:

```
# service httpd start
# chkconfig httpd on
```

4. To use the `netstat` command to see on which ports the `httpd` server is listening, enter the following:

```
# netstat -tupln | grep httpd
tcp6      0      0 :::80          :::*           LISTEN     2496/httpd
tcp6      0      0 :::443         :::*           LISTEN     2496/httpd
```

5. Try to connect to your Apache web server from a web browser that is outside of the local system. If it fails, correct any problems that you encounter by investigating the firewall, SELinux, and other security features.

If you don’t have DNS set up yet, use the IP address of the server to view your Apache server from a remote web browser, such as `http://192.168.0.1`. If you are not able to connect, retry connecting to the server from your browser after performing each of the following steps on the system running the Apache server:

```
# iptables -F
# setenforce 0
# chmod 644 /var/www/html/index.html
```

The `iptables -F` command flushes the firewall rules temporarily. If connecting to the web server succeeds after that, you need to add new firewall rules to open `tcp` ports 80 and 443 on the server. On a system using the `firewalld` service, do this by clicking the check box next to those ports on the Firewall window. For systems running the `iptables` service, add the following rules before the last `DROP` or `REJECT` rule.

```
-A INPUT -m state --state NEW -m tcp -p tcp --dport 80 -j
ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 443 -j
ACCEPT
```

The `setenforce 0` command puts SELinux in permissive mode temporarily. If connecting to the web server succeeds after that, you need to correct SELinux file context and/or Boolean issues (probably file context in this case). The following should work:

```
# chcon --reference=/var/www/html /var/www/html/index.html
```

If the `chmod` command works, it means that the Apache user and group did not have read permission to the file. You should be able to leave the new permissions as they are.

6. To use the `openssl` or similar command to create your own private RSA key and self-signed SSL certificate, do the following:

```
# yum install openssl
# cd /etc/pki/tls/private
# openssl genrsa -out server.key 1024
# chmod 600 server.key
# cd /etc/pki/tls/certs
# openssl req -new -x509 -nodes -sha1 -days 365 \
  -key /etc/pki/tls/private/server.key \
  -out server.crt
Country Name (2 letter code) [AU]: US
State or Province Name (full name) [Some-State]: NJ
Locality Name (eg, city) []: Princeton
Organization Name (eg, company) [Internet Widgits Pty
Ltd]:TEST USE ONLY
Organizational Unit Name (eg, section) []:TEST USE ONLY
Common Name (eg, YOUR name) []:secure.example.org
Email Address []:dom@example.org
```

You should now have a `/etc/pki/tls/private/server.key` key file and a `/etc/pki/tls/certs/server.crt` certificate file.

7. To configure your Apache web server to use your key and self-signed certificate to serve secure (HTTPS) content, do the following:
  - a. Edit the `/etc/httpd/conf.d/ssl.conf` file to change the key and certificate locations to use the ones that you just created:

```
SSLCertificateFile /etc/pki/tls/certs/server.crt
SSLCertificateKeyFile /etc/pki/tls/private/server.key
```

- b. Restart the `httpd` service:

```
# systemctl restart httpd.service
```

8. To use a web browser to create an HTTPS connection to your web server and view the contents of the certificate that you created, do the following:

From the system running the Apache server, type **https://localhost** in the browser's location box. You should see a message that reads, "This Connection is Untrusted." To complete the connection, do the following:

- a. Click I Understand the Risks.
  - b. Click Add Exception.
  - c. Click Get Certificate.
  - d. Click Confirm Security Exception.
9. To create a file named `/etc/httpd/conf.d/example.org.conf`, which turns on name-based virtual hosting and creates a virtual host that (1) listens on port 80 on all interfaces, (2) has a server administrator of `joe@example.org`, (3) has a server name of `joe.example.org`, (4) has a DocumentRoot of `/var/www/html/joe.example.org`, and (5) has a DirectoryIndex that includes at least `index.html` and then create an `index.html` file in DocumentRoot that contains the words "Welcome to the House of Joe" inside, do the following.

Create an `example.org.conf` file that looks like the following:

```
NameVirtualHost *:80
<VirtualHost *:80>
    ServerAdmin      joe@
example.org
    ServerName       joe.
example.org
    ServerAlias       web.example.org
    DocumentRoot     /var/www/html/joe.example.org/
    DirectoryIndex   index.html
</VirtualHost>
```

This is how you could create the text to go into the `index.html` file:

```
# echo "Welcome to the House of Joe" > \
/var/www/html/joe.example.org/index.html
```

10. To add the text `joe.example.org` to the end of the `localhost` entry in your `/etc/hosts` file on the machine that is running the web server, and check it by typing **http://joe.example.org** into the location box of your web browser to see "Welcome to the House of Joe" when the page is displayed, do the following:

- a. Reload the `httpd.conf` file modified in the previous exercise in one of two ways:

```
# apachectl graceful
# systemctl restart httpd
```

- b. Edit the `/etc/hosts` file with any text editor, so the local host line appears as follows:

```
127.0.0.1      localhost.localdomain localhost joe.
example.org
```

- c. From a browser on the local system where `httpd` is running, you should be able to type `http://joe.example.org` into the location box to access the Apache web server using name-based authentication.

## Chapter 18: Configuring an FTP Server

### CAUTION

Don't do the tasks described here on a working, public FTP server, because these tasks will interfere with its operations. (You could, however, use these tasks to set up a new FTP server.)

1. To determine which package provides the Very Secure FTP Daemon service, enter the following as root:

```
# yum search "Very Secure FTP"
...
===== N/S Matched: Very Secure FTP =====
vsftpd.i686 : Very Secure Ftp Daemon
```

The search found the `vsftpd` package.

2. To install the Very Secure FTP Daemon package on your system and search for the configuration files in the `vsftpd` package, enter the following:

```
# yum install vsftpd
# rpm -qc vsftpd | less
```

3. To enable anonymous FTP and disable local user login for the Very Secure FTP Daemon service, set the following in the `/etc/vsftpd/vsftpd.conf` file:

```
anonymous_enable=YES
write_enable=YES
anon_upload_enable=YES
local_enable=NO
```

4. To start the Very Secure FTP Daemon service and set it to start when the system boots, enter the following on a current Fedora or Red Hat Enterprise Linux system:

```
# systemctl start vsftpd.service
# systemctl enable vsftpd.service
```



On a Red Hat Enterprise Linux 6 system, enter the following:

```
# service vsftpd start
# chkconfig vsftpd on
```

5. On the system running your FTP server, enter the following to create a file named `test` in the anonymous FTP directory that contains the words “Welcome to your vsftpd server”:

```
# echo "Welcome to your vsftpd server" > /var/ftp/test
```

6. To open the `test` file from the anonymous FTP home directory using a web browser on the system running your FTP server, do the following.

Open a web browser, enter the following in the location box, and press Enter:

```
ftp://localhost/test
```

The text “Welcome to your vsftpd server” should appear in the browser window.

7. To access the `test` file in the anonymous FTP home directory, do the following. (If you cannot access the file, check that your firewall, SELinux, and TCP wrappers are configured to allow access to that file, as described here.)

- a. Enter the following into the location box of a browser on a system on your network that can reach the FTP server (replace `host` with your system’s fully qualified hostname or IP address):

```
ftp://host/test
```

If you cannot see the welcome message in your browser window, check what may be preventing access. To turn off your firewall temporarily (flush your `iptables` rules), enter the following command as the root user from a shell on your FTP server system and then try to access the site again:

```
# iptables -F
```

- b. To disable SELinux temporarily, enter the following and then try to access the site again:

```
# setenforce 0
```

After you have determined what is causing the file on your FTP server to be unavailable, go back to the section “Securing Your FTP Server” in Chapter 18, and go through the steps to determine what might be blocking access to your file. These are the likely possibilities:

- c. For `iptables`, make sure that there is a rule opening TCP port 21 on the server.
- d. For SELinux, make sure that the file context is set to `public_content_t`.

8. To configure your vsftpd server to allow file uploads by anonymous users to a directory named `in`, do the following as root on your FTP server:
  - a. Create the `in` directory as follows:
 

```
# mkdir /var/ftp/in
# chown ftp:ftp /var/ftp/in
# chmod 777 /var/ftp/in
```
  - b. For a recent Fedora or RHEL, open the Firewall Configuration window and check the FTP box under services to open access to your FTP service. For earlier RHEL and Fedora systems, configure your iptables firewall to allow new requests on TCP port 21 by adding the following rule at some point before a final DROP or REJECT rule in your `/etc/sysconfig/iptables` file:
 

```
-A INPUT -m state --state NEW -m tcp -p tcp --dport 21
-j ACCEPT
```
  - c. Configure your iptables firewall to do connection tracking by loading the appropriate module to the `/etc/sysconfig/iptables-config` file:
 

```
IPTABLES_MODULES="nf_conntrack_ftp"
```
  - d. For SELinux to allow uploading to the directory, first set file contexts properly:
 

```
# semanage fcontext -a -t public_content_rw_t "/var/ftp/
in(/.*)?"
# restorecon -F -R -v /var/ftp/in
```
  - e. Next, set the SELinux Boolean to allow uploading:
 

```
# setsebool -P allow_ftp_anon_write on
```
  - f. Restart the vsftpd service (`service vsftpd restart` or `systemctl restart vsftpd.service`).
9. To install the `lftp` FTP client (if you don't have a second Linux system, install `lftp` on the same host running the FTP server). Optionally, try to upload the `/etc/hosts` file to the `in` directory on the server, to make sure it is accessible. Run the following commands as the root user:

```
# yum install lftp
# lftp localhost
lftp localhost: /> cd in
lftp localhost: /in> put /etc/hosts
89 bytes transferred
lftp localhost: /in> quit
```

You won't be able to see that you copied the `hosts` file to the incoming directory. However, enter the following from a shell on the host running the FTP server to make sure that the `hosts` file is there:

```
# ls /var/ftp/in/hosts
```

If you cannot upload the file, troubleshoot the problem as described in Exercise 7, recheck your `vsftpd.conf` settings, and review the ownership and permissions on the `/var/ftp/in` directory.

10. Using any FTP client you choose, visit the `/pub/debian-meetings` directory on the `ftp://ftp.gnome.org` site and list the contents of that directory. Here's how to do that with the `lftp` client:

```
# lftp ftp://ftp.gnome.org/pub/debian-meetings/
cd ok, cwd=/pub/debian-meetings
lftp ftp.gnome.org:/pub/debian-meetings>> ls
drwxr-xr-x   3 ftp      ftp           3 Jan 13   2014 2004
drwxr-xr-x   6 ftp      ftp           6 Jan 13   2014 2005
drwxr-xr-x   8 ftp      ftp           8 Dec 20   2006 2006
...
```

## Chapter 19: Configuring a Windows File Sharing (Samba) Server

---

1. To install the `samba` and `samba-client` packages, enter the following as root from a shell on the local system:

```
# yum install samba samba-client
```

2. To start and enable the `smb` and `nmb` services, enter the following as root from a shell on the local system:

```
# systemctl enable smb.service
# systemctl start smb.service
# systemctl enable nmb.service
# systemctl start nmb.service
```

or

```
# chkconfig smb on
# service smb start
# chkconfig nmb on
# service nmb start
```

3. To set the Samba server's workgroup to `TESTGROUP`, the NetBIOS name to `MYTEST`, and the server string to `Samba Test System`, as root user in a text editor, open the `/etc/samba/smb.conf` file, and change three lines so that they appear as follows:

```
workgroup = TESTGROUP
netbios name = MYTEST
server string = Samba Test System
```

4. To add a Linux user named `phil` to your system, and add a Linux password and Samba password for `phil`, enter the following as root user from a shell. (Be sure to remember the passwords you set.)

```
# useradd phil
# passwd phil
New password: *****
Retype new password: *****
# smbpasswd -a phil
New SMB password: *****
Retype new SMB password: *****
Added user phil.
```

5. To set the `[homes]` section so that home directories are browseable (yes) and writeable (yes), and that `phil` is the only valid user, open the `/etc/samba/smb.conf` file as root, and change the `[homes]` section so that it appears as follows:

```
[homes]
    comment = Home Directories
    browseable = Yes
    read only = No
    valid users = phil
```

6. To set SELinux Booleans that are necessary to make it so that `phil` can access his home directory via a Samba client, enter the following as root from a shell, and restart the `smb` and `nmb` services:

```
# setsebool -P samba_enable_home_dirs on
# systemctl restart smb
# systemctl restart nmb
```

7. From the local system, use the `smbclient` command to list that the `homes` share is available.

```
# smbclient -L localhost
Enter TESTGROUP\root's password: <ENTER>
Anonymous login successful

      Sharename      Type      Comment
      -----      -
      homes          Disk      Home Directories
      ...
```

8. To connect to the `homes` share from a Nautilus (file manager) window on the Samba server's local system for the user `phil` in a way that allows you to drag and drop files to that folder, do the following:
- Open the Nautilus window (select the files icon).
  - In the left pane, select Other Locations and then click in the Connect to Server box.
  - Type the Server address. For example, `smb://localhost/phil/`.

- d. When prompted, select Registered User, type **phil** as the username, enter the domain (TESTGROUP), and enter phil's password.
  - e. Open another Nautilus window and drop a file to phil's homes folder.
9. To open up the firewall so that anyone who has access to the server can access the Samba service (smbd and nmbd daemons), you can simply open the Firewall Configuration window and check the **samba** and **samba-client** check boxes (for both Runtime and Permanent). If your system is running basic iptables (and not the firewalld service), change the `/etc/sysconfig/iptables` file so that the firewall appears like the following (the rules you add being those in bold):

```
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-I INPUT -m state --state NEW -m udp -p udp --dport 137 -j ACCEPT
-I INPUT -m state --state NEW -m udp -p udp --dport 138 -j ACCEPT
-I INPUT -m state --state NEW -m tcp -p tcp --dport 139 -j ACCEPT
-I INPUT -m state --state NEW -m tcp -p tcp --dport 445 -j ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
COMMIT
```

Then enter the following for the firewall rules to be reloaded:

```
# service iptables restart
```

10. To open the homes share again as the user **phil** from another system on your network (Windows or Linux), and make sure that you can drag and drop files to it, do the following:
- a. This step is really just repeating the Nautilus example described previously or accessing a Windows File Explorer window and opening the share (by selecting Network, then the Samba server). The trick is to make sure that the service has been made available through the Linux server security features.
  - b. If you cannot access the Samba share, try disabling your firewall and then disabling SELinux. If the share is accessible when you turn off either of those services, go back and debug the problems with the service that is not working:

```
# setenforce 0
# service iptables stop
```

- c. When you have fixed the problem, set SELinux back to Enforcing mode and restart iptables:

```
# setenforce 1
# service iptables start
```

## Chapter 20: Configuring an NFS File Server

1. To install the packages needed to configure the NFS service on your chosen Linux system, enter the following as root user at a shell (Fedora or RHEL):

```
# yum install nfs-utils
```

2. To list the documentation files that come in the package that provides the NFS server software, enter the following:

```
# rpm -qd nfs-utils
/usr/share/doc/nfs-utils-1.2.5/ChangeLog
...
/usr/share/man/man5/exports.5.gz
/usr/share/man/man5/nfs.5.gz
/usr/share/man/man5/nfsmount.conf.5.gz
/usr/share/man/man7/nfsd.7.gz
/usr/share/man/man8/blkmapd.8.gz
/usr/share/man/man8/exportfs.8.gz
...
```

3. To start and enable the NFS service, enter the following as root user on the NFS server:

```
# systemctl start nfs-server.service
# systemctl enable nfs-server.service
```

4. To check the status of the NFS service that you just started on the NFS server, enter the following as root user:

```
# systemctl status nfs-server.service
```

5. To share a directory `/var/mystuff` from your NFS server as available to everyone, read-only, and with the root user on the client having root access to the share, first create the mount directory as follows:

```
# mkdir /var/mystuff
```

Then create an entry in the `/etc/exports` file that is similar to the following:

```
/var/mystuff    *(ro,no_root_squash,insecure)
```

To make the share available, enter the following:

```
# exportfs -v -a
exporting */var/mystuff
```

6. To make sure that the share you created is accessible to all hosts, first check that `rpcbind` is not blocked by TCP wrappers by adding the following entry to the beginning of the `/etc/hosts.allow` file:

```
rpcbind: ALL
```

- a. To open the firewall in systems that use `firewalld` (RHEL 8 and recent Fedora systems), install the `firewall-config` package. Then run `firewall-config`. From the Firewall Configuration window that appears, make sure that `nfs` and `rpc-bind` are checked to On for the Permanent firewall settings.
- b. To open the ports needed to allow clients to reach NFS through the `iptables` firewall (RHEL 6 and earlier Fedora systems without `firewalld`), you need to open at least TCP and UDP ports 111 (`rpcbind`), 20048 (`mountd`), and 2049 (`nfs`) by adding the following rules to the `/etc/sysconfig/iptables` file and starting the `iptables` service:

```
-A INPUT -m state --state NEW -m tcp -p tcp --dport 111 -j ACCEPT
-A INPUT -m state --state NEW -m udp -p udp --dport 111 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 2049 -j ACCEPT
-A INPUT -m state --state NEW -m udp -p udp --dport 2049 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 20048 -j ACCEPT
-A INPUT -m state --state NEW -m udp -p udp --dport 20048 -j ACCEPT
```

SELinux should be able to share NFS filesystems while in enforcing mode without any changes to file contexts or Booleans. To make sure that the share you created can be shared read-only, run the following command as root user on the NFS server:

```
# setsebool -P nfs_export_all_ro on
```

7. To view the shares available from the NFS server, assuming that the NFS server is named `nfsserver`, enter the following from the NFS client:

```
# showmount -e nfsserver
Export list for nfsserver:
/var/mystuff *
```

8. To create a directory called `/var/remote` and temporarily mount the `/var/mystuff` directory from the NFS server (named `nfsserver` in this example) on that mount point, enter the following as root user from the NFS client:

```
# mkdir /var/remote
# mount -t nfs nfsserver:/var/mystuff /var/remote
```

9. To add an entry so that the same mount is done automatically when you reboot, first unmount `/var/remote` as follows:

```
# umount /var/remote
```

Then add an entry like the following to the `/etc/fstab` on the client system:

```
/var/remote  nfserver:/var/mystuff  nfs bg,ro 0 0
```

To test that the share is configured properly, enter the following on the NFS client as the root user:

```
# mount -a
# mount -t nfs4
nfserver:/var/mystuff on /var/remote type nfs4
(ro,vers=4,rsize=524288...
```

10. To copy some files to the `/var/mystuff` directory, enter the following on the NFS server:

```
# cp /etc/hosts /etc/services /var/mystuff
```

From the NFS client, to make sure that you can see the files just added to that directory, and to make sure that you can't write files to that directory from the client, enter the following:

```
# ls /var/remote
hosts      services
# touch /var/remote/file1
touch: cannot touch '/var/remote/file1': Read-only file
system
```

## Chapter 21: Troubleshooting Linux

1. To go into Setup mode from the BIOS screen on your computer, do the following:
  - a. Reboot your computer.
  - b. Within a few seconds, you should see the BIOS screen, with an indication of which function key to press to go into Setup mode. (On my Dell workstation, it's the F2 function key.)
  - c. The BIOS screen should appear. (If the system starts booting Linux, you didn't press the function key fast enough.)
2. From the BIOS setup screen, do the following to determine whether your computer is 32-bit or 64-bit, whether it includes virtualization support, and whether your network interface card is capable of PXE booting.

Your experience may be a bit different from mine, depending on your computer and Linux system. The BIOS setup screen is different for different computers. In general, however, you can use arrow keys and tab keys to move between different columns, and press Enter to select an entry.



- a. On my Dell workstation, under the System heading, I highlight Processor Info to see that mine is a 64-bit Technology computer. Look in the Processor Info section, or a similar, section on your computer, to see the type of processor that you have.
  - b. On my Dell workstation, under the Onboard Devices heading, I highlight Integrated NIC and press Enter. The Integrated NIC screen that appears to the right lets me choose to enable or disable the NIC (On or Off) or enable with PXE or RPL (if I intend to boot the computer over the network).
3. To interrupt the boot process to get to the GRUB boot loader, do the following:
  - a. Reboot the computer.
  - b. Just after the BIOS screen disappears, when you see the countdown to booting the Linux system, press any key (perhaps the spacebar).
  - c. The GRUB boot loader menu should appear, ready to allow you to select which operating system kernel to boot.
4. To boot up your computer to runlevel 1 so that you can do some system maintenance, get to the GRUB boot screen (as described in the previous exercise), and then do the following:
  - a. Use the arrow keys to highlight the operating system and kernel that you want to boot.
  - b. Type **e** to see the entries needed to boot the operating system.
  - c. Move your cursor to the line that included the kernel. (It should include the word `vmlinuz` somewhere on the line.)
  - d. Move the cursor to the end of that line, add a space, and then type **init=bash**.
  - e. Follow the instructions to boot the new entry. You will probably either press Ctrl+X or press Enter; if there is another screen, type **b**.

If it worked, your system should bypass the login prompt and boot up directly to a root user shell where you can do administrative tasks without providing a password.
5. To look at the messages that were produced in the kernel ring buffer (which shows the activity of the kernel as it booted up), enter the following from the shell after the system finishes booting:

```
# dmesg | less
```

6. Or, on a system using `systemd`, enter the following:

```
# journalctl -k
```

7. To run a trial `yum update` from Fedora or RHEL and exclude any kernel package that is available, enter the following (when prompted, type **N** to not actually go through with the update, if updates are available):

```
# yum update --exclude='kernel*'
```

8. To check to see what processes are listening for incoming connections on your system, enter the following:

```
# netstat -tupln | less
```

9. To check to see what ports are open on your external network interface, do the following.

If possible, run the `nmap` command from another Linux system on your network, replacing *yourhost* with the hostname or IP address of your system:

```
# nmap yourhost
```

10. To clear your system's page cache and watch the effect it has on your memory usage, do the following:

- Select Terminal from an application menu on your desktop (it is located on different menus for different systems).

- Run the `top` command (to watch processes currently running on your system), and then type a capital **M** to sort processes by those consuming the most memory.

- From the Terminal window, select File and Open Terminal to open a second Terminal window.

- From the second Terminal window, become root user (`su -`).

- While watching the Mem line (used column) in the first Terminal window, enter the following from the second Terminal window:

```
# echo 3 > /proc/sys/vm/drop_caches
```

- The used RES memory should go down significantly on the Mem line. The numbers in the RES column for each process should go down as well.

11. To view memory and swap usage from Cockpit through your web browser, open your browser to Cockpit for your host (<https://hostname:9090>). Then select System ⇄ Memory & Swap.

## Chapter 22: Understanding Basic Linux Security

1. To check log messages from the `systemd` journal for the `NetworkManager.service`, `sshd.service`, and `auditd.service` services, enter the following:

```
# journalctl -u NetworkManager.service
...
```

```
# journalctl -u sshd.service
...
# journalctl -u auditd.service
...
```

2. User passwords are stored in the `/etc/shadow` file. To see its permissions, type `ls -l /etc/shadow` at the command line. (If no shadow file exists, then you need to run `pwconv`.)

The following are the appropriate settings:

```
# ls -l /etc/shadow
-----. 1 root root 1049 Feb 10 09:45 /etc/shadow
```

3. To determine your account's password aging and whether it will expire using a single command, type `chage -l user_name`. For example:

```
# chage -l chris
```

4. To start auditing writes to the `/etc/shadow` with the `auditd` daemon, enter the following at the command line:

```
# auditctl -w /etc/shadow -p w
```

To check your audit settings, type in `auditctl -l` at the command line.

5. To create a report from the `auditd` daemon on the `/etc/shadow` file, enter `ausearch -f /etc/shadow` at the command line. To turn off the auditing on that file, enter `auditctl -W /etc/shadow -p w` at the command line.
6. To install the `lemon` package, damage the `/usr/bin/lemon` file, verify that the file has been tampered with, and remove the `lemon` package, enter the following:

```
# yum install -y lemon
# cp /etc/services /usr/bin/lemon
# rpm -V lemon
S.5...T. /usr/bin/lemon
# yum erase lemon
```

From the original `lemon` file, the file size (S), the `md4sum` (5), and the modification times (T) all differ. For Ubuntu, install the package with `apt-get install lemon` and enter `debsums lemon` to check it.

7. If you suspect that you have had a malicious attack on your system today and important binary files have been modified, you can find these modified files by entering the following at the command line: `find directory -mtime -1` for the directories, `/bin`, `/sbin`, `/usr/bin`, and `/usr/sbin`.
8. To install and run `chkrootkit` to see if the malicious attack from the exercise above installed a rootkit, choose your distribution and do the following:
  - a. To install on a Fedora or RHEL distribution, enter `yum install chkrootkit` at the command line.

- b. To install on an Ubuntu or Debian-based distribution, enter `sudo apt-get install chkrootkit` at the command line.
- c. To run the check, enter `chkrootkit` at the command line and review the results.
- 9. To find files anywhere in the system with the SUID or SGID permission set, enter `find / -perm /6000 -ls` at the command line.
- 10. To install the `aide` package, run the `aide` command to initialize the `aide` database, copy the database to the correct location, and run the `aide` command to check whether any important files on your system have been modified, enter the following.

```
# yum install aide
# aide -i
# cp /var/lib/aide/aide.db.new.gz /var/lib/aide/aide.db.gz
# aide -C
```

To make the output more interesting, you could install the `lemon` package (described in an earlier exercise) before you run `aide -i`, and modify it before running `aide -C` to see how a modified binary looks from `aide`.

## Chapter 23: Understanding Advanced Linux Security

To do the first few exercises, you must have the `gnupg2` package installed. This is not installed by default in Ubuntu, although it is installed for the latest Fedora and RHEL releases.

- 1. To encrypt a file using the `gpg2` utility and a symmetric key, enter the following command. (The `gpg2` utility asks for a passphrase to protect the symmetric key.)

```
$ gpg2 -c filename
```

- 2. To generate a key pair using the `gpg2` utility, enter the following:

```
$ gpg2 --gen-key
```

You must provide the following information:

- a. Your real name and email address
- b. A passphrase for the private key

- 3. To list out the keys you generated, enter the following:

```
$ gpg2 --list-keys
```

- 4. To encrypt a file and add your digital signature using the `gpg2` utility, do the following:

- a. You must have first generated a key ring (Exercise 2).
- b. After you have generated the key ring, enter

```
$ gpg2 --output EncryptedSignedFile --sign
FiletoEncryptSign
```

5. From the `getfedora.org` page, select one of the Fedora distributions to download. When the download is complete, select Verify your Download to see instructions for verifying your image. For example, download the appropriate CHECKSUM file for your image, then enter the following:

```
$ curl https://getfedora.org/static/fedora.gpg | gpg
--import
$ gpg --verify-files *-CHECKSUM
$ sha256sum -c *-CHECKSUM
```

6. To determine if the `su` command on your Linux system is PAM-aware, enter the following:

```
$ ldd $(which su) | grep pam
libpam.so.0 => /lib64/libpam.so.0 (0x00007fca14370000)
libpam_misc.so.0 => /lib64/libpam_misc.so.0
(0x00007fca1416c000)
```

If the `su` command on your Linux system is PAM-aware, you should see a PAM library name listed when you issue the `ldd` command.

7. To determine if the `su` command has a PAM configuration file, type the following:

```
$ ls /etc/pam.d/su
/etc/pam.d/su
```

If the file exists, type the following at the command line to display its contents. The PAM contexts it uses include any of the following: `auth`, `account`, `password`, or `session`.

```
$ cat /etc/pam.d/su
```

8. To list out the various PAM modules on your Fedora or RHEL system, enter the following:

```
$ ls /usr/lib64/security/pam*.so
```

To list out the various PAM modules on your Ubuntu Linux system, enter the following:

```
# find / -name pam*.so
```

9. To find the PAM “other” configuration file on your system, enter `ls /etc/pam.d/other` at the command line. An “other” configuration file that enforces Implicit Deny should look similar to the following code:

```
$ cat /etc/pam.d/other
#%PAM-1.0
auth      required      pam_deny.so
account   required      pam_deny.so
password  required      pam_deny.so
session   required      pam_deny.so
```

10. To find the PAM limits configuration file, enter the following:

```
$ ls /etc/security/limits.conf
```

Display the file's contents by entering the following:

```
$ cat /etc/security/limits.conf
```

Settings in this file to prevent a fork bomb look like the following:

```
@student      hard      nproc          50
@student      -          maxlogins       4
```

## Chapter 24: Enhancing Linux Security with SELinux

1. To set your system into the permissive mode for SELinux, enter `setenforce permissive` at the command line. It would also be acceptable to enter `setenforce 0` at the command line.
2. To set your system into the enforcing operating mode for SELinux without changing the SELinux primary configuration file, use caution. It is best not to run this command on your system for an exercise until you are ready for the SELinux to be enforced. Use the following command at the command line: `setenforce enforcing`. It would also be acceptable to enter `setenforce 1` at the command line.
3. To find and view the permanent SELinux policy type (set at boot time), go to the main SELinux configuration file, `/etc/selinux/config`. To view it, enter `cat /etc/selinux/config | grep SELINUX=` at the command line. To be sure how it is currently set, enter the `getenforce` command.
4. To list the `/etc/hosts` file security context and identify the different security context attributes, enter `ls -Z /etc/hosts` at the command line:

```
$ ls -Z /etc/hosts
-rw-r--r--. root root system_u:object_r:net_conf_t:s0 /etc/hosts
```

- a. The file's user context is `system_u`, indicating a system file.
- b. The file's role is `object_r`, indicating an object in the file system (a text file, in this case).
- c. The file's type is `net_conf_t`, because the file is a network configuration file.
- d. The file's sensitivity level is `s0`, indicating the lowest security level. (This number may be listed in a range of numbers from `s0-s3`.)
- e. The file's category level starts with a `c` and ends with a number. It may be listed in a range of numbers, such as `c0-c102`. This is not required except in highly secure environments and is not set here.

5. To create a file called `test.html` and assign its type as `httpd_sys_content_t`, enter the following:

```
$ touch test.html
$ chcon -t httpd_sys_content_t test.html
$ ls -Z test.html
-rw-rw-r--. chris chris unconfined_u:object_r:httpd_sys_
content_t:s0 test.html
```

6. To list the `crond` process's security context and identify the different security context attributes, enter this at the command line:

```
$ ps -efZ | grep crond
system_u:system_r:crond_t:s0-s0:c0.c1023 root 665 1 0
Sep18 ? 00:00:00 /usr/sbin/crond -n
```

- a. The process's user context is `system_u`, indicating a system process.
  - b. The process's role is `system_r`, indicating a system role.
  - c. The process's type or domain is `crond_t`.
  - d. The process's sensitivity level starts `s0-s0`, indicating that it is not highly sensitive. (It is secure by normal Linux standards, however, because the process is run as the root user.)
  - e. The process's category level is `c0.c1023`, with the `c0`, indicating that the category is also not highly secure from an SELinux standpoint.
7. To create an `/etc/test.txt` file, change its file context to `user_tmp_t`, restore it to its proper content (the default context for the `/etc` directory), and remove the file, enter the following:

```
# touch /etc/test.txt
# ls -Z /etc/test.txt
-rw-r--r--. root root unconfined_u:object_r:etc_t:s0 /etc/
test.txt
# chcon -t user_tmp_t /etc/test.txt
# ls -Z /etc/test.txt
-rw-r--r--. root root unconfined_u:object_r:user_tmp_t:s0 /
etc/test.txt
# restorecon /etc/test.txt
# ls -Z /etc/test.txt
-rw-r--r--. root root unconfined_u:object_r:etc_t:s0 /etc/
test.txt
# rm /etc/test.txt
rm: remove regular empty file `/etc/test.txt'? y
```

8. To determine what Booleans allow anonymous writes and access to the `tftp` service's home directory, then turn those Booleans on permanently, enter the following commands:

```
# getsebool -a | grep tftp
tftp_home_dir --> off
tftpd_anon_write --> off
```

```
...
# setsebool -P tftp_home_dir=on
# setsebool -P tftp_anon_write=on
# getsebool tftp_home_dir tftp_anon_write
tftp_home_dir --> on
tftp_anon_write --> on
```

9. To list all SELinux policy modules on your system, along with their version numbers, enter **semodule -l**.

### NOTE

If you wrote `ls /etc/selinux/targeted/modules/active/modules/*.pp` as your answer, that is okay, but this command doesn't give you the version numbers of the policy modules. Only `semodule -l` gives the version numbers.

10. To tell SELinux to allow access to the `sshd` service through TCP Port 54903, enter the following:

```
# semanage port -a -t ssh_port_t -p tcp 54903
# semanage port -l | grep ssh
ssh_port_t                                tcp                                54903, 22
```

## Chapter 25: Securing Linux on a Network

1. To install the Network Mapper (aka `nmap`) utility on your local Linux system:
  - a. On Fedora or RHEL, enter `yum install nmap` at the command line.
  - b. On Ubuntu, `nmap` may come pre-installed. If not, enter `sudo apt-get install nmap` at the command line.
2. To run a TCP Connect scan on your local loopback address, enter `nmap -sT 127.0.0.1` at the command line. The ports you have running on your Linux server will vary. However, they may look similar to the following:

```
# nmap -sT 127.0.0.1
...
PORT      STATE SERVICE
25/tcp    open  smtp
631/tcp    open  ipp
```

3. To run a UDP Connect scan on your Linux system from a remote system:
  - a. Determine your Linux server's IP address by entering **ifconfig** at the command line. The output will look similar to the following, and your system's IP address follows `inet addr:` in the `ifconfig` command's output.

```
# ifconfig
...
p2p1  Link encap:Ethernet  HWaddr 08:00:27:E5:89:5A
      inet addr:10.140.67.23
```



- b. From a remote Linux system, enter the command `nmap -sU IP address` at the command line, using the *IP address* you obtained from above.  
For example:  

```
# nmap -sU 10.140.67.23
```
4. To check to see if your system is running the `firewalld` service, and then install and start it if it is not:
  - a. Enter `systemctl status firewalld.service`.
  - b. If the `firewalld` service is not running, on a Fedora or RHEL system, enter the following:  

```
# yum install firewalld firewall-config -y
# systemctl start firewalld
# systemctl enable firewalld
```
5. To open ports in your firewall to allow remote access to your local web service, do the following:
  - a. Start the Firewall Configuration window (`firewalld-config`).
  - b. Make sure that Configuration: Runtime is selected.
  - c. Select your current zone (for example, `FedoraWorkstation`).
  - d. Under Services, select the `http` and `https` check boxes.
  - e. Select Configuration: Permanent.
  - f. Under Services, select the `http` and `https` check boxes.
6. To determine your Linux system's current `netfilter/iptables` firewall policies and rules, enter `iptables -vnl` at the command line.
7. To save, flush, and restore your Linux system's current firewall rules:
  - a. To save your current rules:  

```
# iptables-save >/tmp/myiptables
```
  - b. To flush your current rules:  

```
# iptables -F
```
  - c. To restore the firewall's rules, enter:  

```
# iptables-restore < /tmp/myiptables
```
8. To set your Linux system's firewall filter table for the input chain to a policy of `DROP`, enter `iptables -P INPUT DROP` at the command line.
9. To change your Linux system firewall's filter table policy back to `accept` for the input chain, enter the following:  

```
# iptables -P INPUT ACCEPT
```

To add a rule to drop all network packets from the IP address 10.140.67.23, enter the following:

```
# iptables -A INPUT -s 10.140.67.23 -j DROP
```

10. To remove the rule that you just added, without flushing or restoring your Linux system firewall's rules, enter `iptables -D INPUT 1` at the command line. This is assuming that the rule you added above is rule 1. If not, change the 1 to the appropriate rule number in your `iptables` command.

## Chapter 26: Shifting to Clouds and Containers

1. To install and start either `podman` (for any RHEL or Fedora system) or `docker` (RHEL 7):

```
# yum install podman -y
or
# yum install docker -y
# systemctl start docker
# systemctl enable docker
```

2. To use either `docker` or `podman` to pull this image to your host, `registry.access.redhat.com/ubi7/ubi`:

```
# podman pull registry.access.redhat.com/ubi7/ubi
or
# docker pull registry.access.redhat.com/ubi7/ubi
```

3. To run the `ubi7/ubi` image to open a bash shell:

```
# podman run -it ubi7/ubi bash
or
# docker run -it ubi7/ubi bash
```

4. To run commands to see the operating system on which the container is based, install the `proc-ps` package, and run a command to see the processes running inside the container:

```
bash-4.4# cat /etc/os-release | grep ^NAME
NAME="Red Hat Enterprise Linux"
bash-4.4# yum install procps -y
bash-4.4# ps -ef
UID          PID    PPID  C STIME TTY          TIME CMD
root           1         0  0 03:37 pts/0    00:00:00 bash
root          20         1  0 03:43 pts/0    00:00:00 ps -ef
bash-4.4# exit
```

5. To restart and connect to the container that you just closed using an interactive shell, enter the following:

```
# podman ps -a
CONTAINER ID    IMAGE                                COMMAND          CREATED
```

```
STATUS                                PORTS  NAMES
eabf1fb57a3a  ...ubi8/ubi:latest bash      7 minutes ago
Exited (0) 4 seconds ago              compassionate_hawking
# podman start -a eabf1fb57a3a
bash-4.4# exit
```

6. To create a simple Dockerfile from a ubi7/ubi base image, include a script named `cworks.sh` that echoes "The Container Works!", and add that script to the image so that it runs, do the following:

- a. Create and change to a new directory:

```
# mkdir project
# cd project
```

- b. Create a file named `Dockerfile` with the following content:

```
FROM registry.access.redhat.com/ubi7/ubi-minimal
COPY ./cworks.sh /usr/local/bin/
CMD ["/usr/local/bin/cworks.sh"]
```

- c. Create a file named `cworks.sh` with the following content:

```
#!/bin/bash
set -o errexit
set -o nounset
set -o pipefail
echo "The Container Works!"
```

7. Use `docker` or `podman` to build an image named `containerworks` from the `Dockerfile` that you just created.

```
# podman build -t myproject .
or
# docker build -t myproject .
```

8. To gain access to a container registry, either by installing the `docker-distribution` package or getting an account on Quay.io or Docker Hub:

```
# yum install docker-distribution -y
# systemctl start docker-distribution
# systemctl enable docker-distribution
```

or get an account from Quay.io (<https://quay.io/plans/>) or Docker Hub, then:

```
# podman login quay.io
Username: <username>
Password: *****
```

9. To tag and push a new image to a chosen container registry:

```
# podman tag aa0274872f23 \
quay.io/<user>/<imagename>:v1.0
# podman push \
quay.io/<user>/<imagename>:v1.0
```

## Chapter 27: Using Linux for Cloud Computing

1. To check your computer to see if it can support KVM virtualization, enter the following:

```
# cat /proc/cpuinfo | grep --color -E "vmx|svm|lm"
flags      : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr
pge mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm
pbe syscall nx pdpe1gb rdtscp lm constant_tsc arch_perfmon pebs
bts rep_good xtopology nonstop_tsc aperfmperf pni pclmulqdq dtes64
monitor ds_cpl vmx smx es...
...
```

The CPU must support either `vmx` or `svm`. The `lm` indicates that it is a 64-bit computer.

2. To install a Linux system along with the packages needed to use it as a KVM host and, to run the Virtual Machine Manager application, do the following:
  - a. Get a live or installation image from a Linux site (such as [getfedora.org](http://getfedora.org)), and burn it to a DVD (or otherwise make it available to install).
  - b. Boot the installation image, and select to install it to a hard drive.
  - c. For a Fedora Workstation, after the install is complete and you have rebooted, install the following package (for different Linux distributions, you might need to install a package that provides `libvirt` as well):

```
# yum install virt-manager libvirt-daemon-config-network
```

3. To make sure that the `sshd` and `libvirtd` services are running on the system, enter the following:

```
# systemctl start sshd.service
# systemctl enable sshd.service
# systemctl start libvirtd.service
# systemctl enable libvirtd.service
```

4. Get a Linux installation ISO image that is compatible with your hypervisor, and copy it to the default directory used by Virtual Machine Manager to store images. For example, if the Fedora Workstation DVD is in the current directory, you can enter the following:

```
# cp Fedora-Workstation-Live-x86_64-30-1.2.iso /var/lib/
libvirt/images/
```

5. To check the settings on the default network bridge (`virbr0`), enter the following:

```
# ip addr show virbr0
4: virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc
noqueue state UP group default
link/ether de:21:23:0e:2b:c1 brd ff:ff:ff:ff:ff:ff
inet 192.168.122.1/24 brd 192.168.122.255 scope global virbr0
valid_lft forever preferred_lft forever6.
```

6. To install a virtual machine using the ISO image you copied earlier, do the following.
  - a. Enter this command:

```
# virt-manager &
```
  - b. Select File and then select New Virtual Machine.
  - c. Select Local Install Media and click Forward.
  - d. Select Browse, choose the live or install ISO, click Choose Volume, and click Forward.
  - e. Select memory and CPUs and click Forward.
  - f. Select the size of disk that you want to use and click Forward.
  - g. Select “Virtual network default: NAT” (it may already be selected).
  - h. If it all looks okay, click Finish.
  - i. Follow the installation process indicated by the installation ISO.
7. To make sure that you can log in to and use the virtual machine, do the following:
  - a. Double-click the entry for the new virtual machine.
  - b. When the viewer window appears, log in as you would normally.
8. To check that your virtual machine can connect to the Internet or other network outside of the hypervisor, do one of the following:
  - a. Open a web browser, and try to connect to a website on the Internet.
  - b. Open a Terminal window, enter `ping redhat.com`, and then press Ctrl+C to exit.
9. To stop the virtual machine so that it is no longer running:
  - a. Right-click the entry for the VM in the virt-manager window.
  - b. Select Shut Down, and then select Shut Down again.
  - c. If the VM doesn’t shut down immediately, you can select Force Off instead, but that is like pulling the plug out and risks data loss.
10. Start the virtual machine again so that it is running and available:
  - a. Right-click the entry for the VM in the virt-manager window.
  - b. Click Run.

## Chapter 28: Deploying Linux to the Cloud

1. To install the `genisoimage`, `cloud-init`, `qemu-img`, and `virt-viewer` packages, enter:

```
# dnf install genisoimage cloud-init qemu-img virt-viewer
```

2. To obtain a Fedora cloud image, go to <https://getfedora.org/en/cloud/download/>, and download a `qcow2` image. There is one listed with OpenStack named `Fedora-Cloud-Base-31-1.9.x86_64.qcow2`.
3. To create a snapshot of that image in `qcow2` format called `myvm.qcow2`, enter the following:

```
# qemu-img create -f qcow2 \
-o backing_file=Fedora-Cloud-Base-31-1.9.x86_64.qcow2 \
myvm.qcow2
```

4. Create a `cloud-init` meta data file named `meta-data` that includes the following content:

```
instance-id: myvm
local-hostname: myvm.example.com
```

5. Create a `cloud-init` user data file called `user-data` that includes the following content:

```
#cloud-config
password: test
chpasswd: {expire: False}
```

6. Run the `genisoimage` command to combine the `meta-data` and `user-data` files to create a `mydata.iso` file:

```
# genisoimage -output mydata.iso -volid cidata \
-joliet-long -rock user-data meta-data
```

7. Use the `virt-install` command to combine the `myvm.qcow2` virtual machine image with the `mydata.iso` image to create a new virtual machine image named `newvm` that runs as a virtual machine on your hypervisor.

```
# virt-install --import --name newvm \
--ram 4096 --vcpus 2 \
--disk path=myvm.qcow2,format=qcow2,bus=virtio \
--disk path=mydata.iso,device=cdrom \
--network network=default &
```

8. To open the `newvm` virtual machine with `virt-viewer`, enter the following:

```
# virt-viewer newvm
```

9. Log into the `newvm` virtual machine using the `fedora` user and password `test`:

```
Login: fedora
Password: test
```

## Chapter 29: Automating Apps and Infrastructure with Ansible

---

1. To install the ansible package, do the following:

### RHEL 8

```
# subscription-manager repos \
    --enable ansible-2.9-for-rhel-8-x86_64-rpms
# dnf install ansible -y
```

### Fedora

```
# dnf install ansible -y
```

### Ubuntu

```
$ sudo apt update
$ sudo apt install software-properties-common
$ sudo apt-add-repository --yes --update ppa:ansible/ansible
$ sudo apt install ansible
```

2. To add sudo privileges for the user running Ansible commands, run visudo and create an entry similar to the following (changing joe to your user name):

```
joe    ALL=(ALL)        NOPASSWD: ALL
```

3. Open a file named my\_playbook.yaml, and add the following content:

```
---
- name: Create web server
  hosts: localhost
  tasks:
    - name: Install httpd
      yum:
        name: httpd
        state: present
```

4. To run the my\_playbook.yaml playbook in check mode, do the following. (It should fail because the user does not have privilege to install a package.)

```
$ ansible-playbook -C my_playbook.yaml
...

TASK [Install httpd]
*****
fatal: [localhost]: FAILED! => {"changed": false, "msg":
"This
    command has to be run under the root user.",
"results": []}
...
```

5. Make the following changes to the `my_playbook.yaml` file:

```
---
- name: Create web server
  hosts: localhost
  become: yes
  become_method: sudo
  become_user: root
  tasks:
  - name: Install httpd
    yum:
      name: httpd
      state: present
```

6. To run the `my_playbook.yaml` file again to install the `httpd` package, enter the following:

```
$ ansible-playbook my_playbook.yaml
...
TASK [Install httpd] *****
changed: [localhost]
PLAY RECAP *****
localhost: ok=2 changed=1 unreachable=0 failed=0 skipped=0
rescued=0 ignored=0
```

7. Modify `my_playbook.yaml` as follows to start the `httpd` service, and set it so that it will start every time the system boots:

```
---
- name: Create web server
  hosts: localhost
  become: yes
  become_method: sudo
  become_user: root
  tasks:
  - name: Install httpd
    yum:
      name: httpd
      state: present
  - name: start httpd
    service:
      name: httpd
      state: started
```

8. To run an `ansible` command so that it checks whether or not the `httpd` service is up on `localhost`, enter the following:

```
$ ansible localhost -m service \
  -a "name=httpd state=started" --check
localhost | SUCCESS => {
```



```
"changed": false,  
"name": "httpd",  
"state": "started",  
"status": { ...
```

9. To create an `index.html` file in the current directory that contains the text "Web server is up" and runs the `ansible` command to copy that file to the `/var/www/html` directory on `localhost`, do the following (changing `joe` to your user name):

```
$ echo "Web server is up" > index.html  
$ ansible localhost  
-m copy -a \  
  "src=./index.html dest=/var/www/html/ \  
  owner=apache group=apache mode=0644" \  
  -b --user joe --become-user root --become-method sudo  
host01 | CHANGED => { ...
```

10. To use the `curl` command to view the contents of the file you just copied to the web server, do the following:

```
$ curl localhost  
Web server is up
```

## Chapter 30: Deploying Applications as Containers with Kubernetes

---

1. To gain access to a Minikube instance, either:
  - a. Install Minikube as described here: <https://kubernetes.io/docs/tasks/tools/install-minikube>, or
  - b. Access an available remote Minikube instance, such as through the Kubernetes.io tutorials: <https://kubernetes.io/docs/tutorials/>

2. To view the versions of your Minikube installation, `kubectl` client, and Kubernetes service, enter the following:

```
$ minikube version  
$ kubectl version
```

3. To create a deployment that manages a pod running the `hello-node` container image, enter the following:

```
$ kubectl create deployment hello-node \  
  --image=gcr.io/hello-minikube-zero-install/hello-node
```

4. To view the `hello-node` deployment and describe the deployment in detail, enter the following:

```
$ kubectl get deployment  
$ kubectl describe deployment hello-node
```

5. To view the current replica set associated with your hello-node deployment, enter the following:

```
$ kubectl get rs
```

6. To scale up the hello-node deployment to three (3) replicas, enter the following:

```
$ kubectl scale deployments/hello-node --replicas=3
```

7. To expose the hello-node deployment outside of the Kubernetes cluster using LoadBalancer, enter the following:

```
$ kubectl expose deployment hello-node \
  --type=LoadBalancer --port=8080
```

8. To get the IP address of your Minikube instance and port number of the exposed hello-node service, enter the following:

```
$ minikube ip
192.168.39.150
$ kubectl describe service hello-node | grep NodePort
NodePort:                <unset> 31302/TCP
```

9. Use the curl command to query the hello-node service, using the IP address and port number from the previous step. For example:

```
$ curl 192.168.39.105:31302
Hello World!
```

10. To delete the hello-node service and deployment and then stop the Minikube virtual machine, enter the following:

```
$ kubectl delete service hello-node
$ kubectl delete deployment hello-node
$ minikube stop
```

# Index

- ( (left parenthesis), 78
- < (less than), 78
- ` (backtick), 80
- ? (question mark), 99
- & (ampersand character), 78
- ; (semicolon), 78
- ) (right parenthesis), 78
- | (pipe character), 78–79
- ;(semicolon), 79
- { } (curly braces), 101
- ~ (tilde) in commands, 97
- # shell prompt, 63
- \$ shell prompt, 63
- > (greater than), 78
- 3D desktop effects, AIGLX, 54–57

## A

- absolute path, 96
- ACLs (Access Control Lists)
  - directories, 267
  - enabling, 265–266
  - restricted deletion directory, 268–269
  - set GID directory, 267–268
  - setfacl command, 262–264
  - setting, 262–264
    - default, 264–265
- AD (Active Directory), 712
- ad-hoc commands, Ansible, 760–762
- administrative commands
  - /bin directory, 179
  - /sbin, 178
  - /usr/bin directory, 179
  - /usr/sbin, 179
- administrative privileges, 168
- administrative utilities, 4–5
- AIGLX (Accelerated Indirect GLX), 54–57
- Alcatel-Lucent, 9
- alias command, 81–83
- aliases, 71
  - completion, 75
  - creating, 81–83
    - network interfaces, 360–361
      - using, 81–83
- aliases file, 181
- amanda, 566
- Amazon EC2, cloud images, 744–746
- ampersand character (&), 78
- anaconda, Linux installation, 205
- Anaconda installer, 17
- anonymous FTP server, 456
- Ansible, 749
  - ad-hoc commands, 760–762
  - applications, deployment, 749
  - containers, 749–750
  - deployment
    - prerequisites, 754
    - SSH keys, 754–755
  - host systems, configuring, 749
  - infrastructure, 749
  - installation
    - authentication, 757
    - inventories, 756–757
    - playbook creation, 757–758
    - playbook running, 758–760
  - inventories, 751–752
  - operators, 749–750
  - playbooks, 749
    - imports, 753
    - includes, 753
    - modules, 752–753
    - plays, 752
    - roles, 753
    - tasks, 752
- Ansible Tower, 762–763
- antivirus software, 591
- Apache HTTPD web server, 427–428, 432–433
  - access denied error, 452
  - configuration files
    - default settings, 438–440
    - directives, 435–436, 435–438
  - httpd package, 428–431
  - index not found error, 452
  - installing, 431

### Apache HTTPD web server (*continued*)

- security
  - file ownership, 433
  - file permissions, 433
  - firewalls, 433–434
  - SELinux and, 434–435
- SSL/TLS and, 443–445
  - certificate signing request, 448–449
  - self-signed certificates, 447–448
  - SSL configuration, 445–447
  - SSL key generation, 447–448
- troubleshooting
  - configuration errors, 449–451
  - errors, accessing, 451–452
  - user published content, 442–443
  - virtual hosts, 440–442
- applets, GNOME, 51
- application-layer firewalls, 674
- applications
  - Ansible, 749–750
  - deploying as containers, 765–783
- aptitude command, 226
- arguments, commands, 67–68
- arithmetic
  - expressions, expanding, 80
  - integer arithmetic, 152–153
  - shell scripts, 152–153
- ASF (Apache Software Foundation), 428
- ash shell, 61
- AT&T, 8
- audits, 595–596
  - network services, 663–665
  - nmap utility, 665–672
- ports, 666
- authentication, 4
  - cloud, 712
  - key-based, Secure Shell tools, 324–326
- PAM
  - contexts, 619–620
  - control flags, 620–621
  - modules, 621–622
  - public key, 313
- author rights, GPL and, 12
- autoofs
  - home directory automount, 518–520
  - /net directory, 517–518
- automation, Ansible Tower and, 762–763
- AWS (Amazon Web Services), 694

## B

- background commands, 79
- background processes, 137–140
- backtick ( ` ), 80
- backup script, 162
- backup utilities, 566
- base image, 696
  - container image, 694
- bash shell, 61, 148
  - configuration files, 84
  - cut command, 159
  - grep, 159
  - prompt, characters, 86
  - sed command, 160
  - text manipulation, 159–161
  - tr command, 160
  - variables, untyped, 152–153
- bashrc file, 181
- Bell labs, 7–8
- Bell-LaPadula Mandatory Access security model, 639
- Berkeley distribution, 9–10
- bigcommand process, 141
- /bin directory, 94
- BIOS (Basic Input Output System), 526–528
- bond0 interface, 362
- bonded network interfaces, 362
- Booleans, SELinux, 653–654
- /boot directory, 94
- boot loaders
  - GRUB (GRand Unified Bootloader), 528–530
    - installation, 217–218
    - troubleshooting, 528–530
  - GRUB 2, 530–531
    - troubleshooting, 530–531
- boot options
  - feature disable, 210
  - kickstarts, 211–212
  - mediacheck, 212
  - rescue mode, 212
  - special installation, 210–211
  - video problems, 210
- boot order, troubleshooting, 527–528
- boot up, 523–524
  - from firmware, 526–528
  - GRUB 2 boot loader, 530–531
  - GRUB boot loader, 528–530

- kernel startup, 532–541
- startup methods, 524
  - init facility, 524–525
  - systemd facility, 525
- from USB drive, 791–792
- bounties, software, 21
- Bourne, Stephen, 61
- Bourne shell, 61
- brace expansion metacharacters, 101
- browser-based admin tools, 173
- BSD (Berkeley Software Distribution), 10, 12–13
  - FreeBSD, 13
  - init daemon, 371–377
  - NetBSD, 13
  - OpenBSD, 13
- built-in commands, 71

## C

- C programming language, 9
- Caesar Cipher, 602
- case command, 156–157
- cd command, 96–97
- CDs/DVDs, burning, 792–795
- Ceph, 5
- certification, 21–22
  - RHCE (Red Hat Certified Engineer), 21, 22
    - topics, 23–25
  - RHCSA (Red Hat Certified System Administrator), 21, 22
    - topics, 22–23
- cgroups, 143–144, 695
- chage command, 72
- chkconfig command, 385–386
- chkrootkit, 592
- chmod command, 100, 106–108, 162
- chronyd package, 309
- CIFS (Common Internet File System), 475–476
- CISA (Cybersecurity and Infrastructure Security Agency), 596
- classes
  - implicit, 404
  - printer classes, 404, 408
- classification level, 639
- cloning, cloud instances, 734–738
- cloud, 3–4
  - hybrid, 731
  - hypervisors, 710, 713
  - configuring, 715–718
    - setup, 714
  - images
    - Amazon EC2, 744–746
    - OpenStack and, 739–744
  - instances, 730
    - cloning, 734–738
    - investigating, 733–734
  - networking, setup, 714
  - platforms, 712
  - private, 730
  - public, 730
  - storage, 711
    - configuring, 718–720
      - setup, 714
    - shared, 713
  - virtual machines, 713
    - creating, 720–724
- cloud computing, 5
  - authentication, 712
  - configuration, 712
  - controllers, 711
  - deployment, 712
- cloud-based installations, 204–205
- cloud-init, 730
  - configuring, 731–733
  - enterprise computing, 738
  - running, 731–733
- clustering, 5
- cnegus-test-project, 739
- Cockpit, 168, 169–171, 249–252
  - firewall rules, 677–678
  - storage management and, 301–303
- command languages, shell and, 62
- command-line
  - argument, 148, 150–151
  - completion, 75–76
  - editing, 73–75
  - network configuration, `nmtui` command, 354
  - NetworkManager TUI, editing connection, 354–355
  - recall, 76–78
- commands, 418
  - ' (backtick), 80
  - ~ (tilde), 97
  - ad-hoc (Ansible), 760–762
  - administrative
    - /sbin, 178
    - /usr/sbin, 179

### commands (*continued*)

- alias, 81–83
  - aliases, 71
  - aptitude, 226
  - arguments, 67–68
  - background, 79
  - built-in, 71
  - case, 156–157
  - cd, 96–97
  - chage, 72
  - chkconfig, 385–386
  - chmod, 100, 106–108, 162
  - completion, 75
  - connecting, 78–81
  - cp, 110
  - cryptsetup, 612
  - cut, 159
  - date, 66
  - df, 334
  - du, 334–335
  - expanding, 78, 80
  - exportfs, 507
  - files, 96–98
  - filesystem, 71
  - find, 122–128, 335–336
  - firewall-config, 674–675
  - functions, 71
  - gedit, 113–114
  - grep, 128–129
  - groupadd, 260–261
  - help, 88
  - here text, 100
  - history, 72–78
  - history, 72–78
  - id, 69
  - info, 89
  - information about, 88–90
  - journalctl, 184
  - kill, 140
  - killall, 140, 141–142
  - lftp, 470–472
  - locate, 72, 120–122
  - locating, 70–72
  - lp, 419
  - lprm, 419–420
  - lpstat, 419
  - ls, 67–68, 101–105
  - man, 89
  - mkfs, 300
  - mount, 297–298
  - mv, 109–110
  - nice, 142–143
  - nmtui, 354
  - one-command actions, 156
  - options, 67–68
  - path, 70
  - pipe ( | )metacharacter, 78–79
  - podman, 694, 697
  - ps, 132–134
  - pwd, 67
  - renice, 142–143
  - reserved words, 71
  - rm, 110
  - rpm, 241–245
  - running, 66–72
  - sar, 332–333
  - secon, 648–649
  - sed, 160
  - sequential, 79
  - setfacl, 262–264
  - sftp, 324
  - ssh, 316
  - su, 168, 175–176
  - sudo, 168
  - syntax, 67–70
  - systemctl, 381
  - telinit, 374
  - text formatting, 79
  - top, 134–135
  - touch, 98–99
  - tr, 160
  - type, 71
  - umount, 299
  - useradd, 252–255
  - userdel, 258–259
  - usermod, 257–258
  - virsh, 711
  - virt-install, 720–721
  - virt-manager, 714
  - virt-viewer, 714
  - who am i, 65
  - yum, 229–232, 233–241
- Compiz, 55
- compliance reviews, 595–596
- compute nodes, 710. *See also* cloud hypervisors
- configuration
- cloud, 712
    - hypervisors, 715–718
    - storage, 718–720
  - files, 310

- administrative, 179–185
    - plain-text files, 179
    - security, 314
  - servers, 310–311
    - configuration files, 310
    - default configuration, 310–311
  - connecting commands, 78–81
  - connectionless protocols, UDP, 666
  - container registries, 694, 695–696
    - images
      - pushing to, 705–706
      - tagging, 705–706
  - containers, 693–694
    - Ansible, 749–750
    - in enterprise, 706
    - FTP, GitHub and, 703–705
    - images, 694
      - base image, 694, 696
      - building, 702–703
      - RPM packages and, 246
    - namespaces, 694, 695
    - pulling, 697–698
    - running FTP servers, 699–701
    - running shells, 698–699
    - sidecar, 766
    - starting/stopping, 701–702
  - control plane, 336
  - copying files, 110
    - interactive copying, 324
    - scp command, 321–324
  - copyrights, GPL and, 12
  - cp command, 110
  - cpio, 566
  - CPU (computer processing unit), 273
  - cracklib, 571
  - cron, software updates and, 545
  - crontab file, 182
  - cryptographic ciphers, 602–603
  - cryptography, 599–600
    - asymmetric keys, 604–605
    - block ciphers, 600
    - cipher keys, 603–608
    - ciphers, 600
    - decryption, 600
    - digital signatures, 608–610
    - email message encryption, 607–608
    - encryption/decryption
      - cipher keys, 603–608
      - ciphers, 602–603
      - digital signatures, 608–610
    - hashing, 600–601
    - implementing
      - directories, 613–615
      - encryption from desktop, 617–618
      - file encryption, 616
      - file integrity, 610–611
      - filesystem encryption, installation, 611–613
      - tools, 616–617
    - key pair generation, 605–606
    - public key sharing, 607
    - stream ciphers, 600
    - symmetric keys, 603–604
    - tar archive files, 604
    - tools, 617
  - cryptsetup command, 612
  - csh (C shell), 61, 65
  - csh.cshrc file, 182
  - CUPS (Common UNIX Printing System), 403–404
    - configuring
      - from a browser, 404
      - manual, 404, 417–418
    - printer drivers, 404
    - printers, adding automatically, 405–406
    - printing to from Windows, 405
    - remote printers, 413
    - server
      - configuring, 415–416
      - starting, 417
    - shared printers, 420–422
    - web-based administration, 406
      - automatic detection, 407–408
      - remote administration, 406–407
  - curly braces {}, 101
  - cut command, 159
  - cutting text, 159
  - CVE (Common Vulnerabilities and Exposures), 580
- ## D
- DAC (Discretionary Access Control), 635–636
  - daemon processes, 5, 179, 307. *See also* services
    - apache, 185
    - avahi, 185
    - bin, 185
    - chrony, 185
    - configuration files, 311
    - lp, 185
    - news, 186
    - permissions and, 311
    - port numbers, 311

- daemon processes (*continued*)
  - postfix, 185
  - rpc, 186
  - services, 369
- dash shell, 61, 65
- datasources, 738
- date command, 66
- DEB packaging, 225
  - Ubuntu Software Center, 225
- Debian, 19
- debugging, shell scripts, 148
- dependencies, 369
- dependent software, 224
- deployment, automatic, 336
- desktop. *See also* GNOME; GNOME 2; GNOME 3; X Window System
  - 3D effects, AIGLX, 54–57
  - GNOME, 29, 30
  - GNOME 3, 31
  - KDE (K Desktop Environment), 29
  - LXDE (Lightweight X11 Desktop Environment), 29
  - window manager, 29
  - Xfce, 29
- desktop networking
  - configuring, NetworkManager, 340–342
  - NetworkManager, 340–342
- /dev directory, 94
- df command, 334
- DHCP, 340–341
- digital signatures, 608–610
- directories
  - /bin, 94
  - /boot, 94
  - /dev, 94
  - encrypting, 613–615
  - /etc, 94, 180
  - /etc/cron, 180
  - /etc/cups, 180, 405
  - /etc/default, 180
  - /etc/exports, 504, 505
  - /etc/httpd, 181
  - /etc/mail, 181
  - /etc/postfix, 181
  - /etc/ppp, 181
  - /etc/rc?.d, 181
  - /etc/security, 181
  - /etc/skel, 181
  - /etc/sysconfig, 181
  - /etc/systemd, 181
  - /etc/X11, 183
  - /etc/xinetd.d, 181
  - hard drive partitions, 216
  - hierarchy, 94
  - /home, 94
  - identifying, 104
  - /lib, 94
  - listing, 101–105
  - /media, 94
  - /misc, 94
  - /mnt, 94
  - number of characters, 103
  - /opt, 94
  - paths, 70
    - absolute path, 96
    - order, 71
  - /proc, 95
  - restricted deletion directory, 268–269
  - root, 93
  - /root, 95
  - /sbin, 95
  - /sys, 95
  - time and date column, 103
  - /tmp, 95
  - /usr, 95
  - /var, 95
- disaster recovery, security, 566
- disk images, mounting in loopback, 298–299
- disk space, 197
- disk storage, 273
- distributions
  - components, 16
  - DEB packaging, 225
  - Debian, 19
  - Fedora, 18–19
  - GPL and, 12
  - Red Hat
    - Anaconda installer, 17
    - graphical administration, 17
    - RPM package management, 16–17
  - Red Hat OpenShift, 18
  - Red Hat OpenStack Platform, 18
  - RHEL (Red Hat Enterprise Linux), 17–18
  - RPM packaging, 225
  - Ubuntu, 19
- DNF (Dandified YUM), 229
- Docker, 697
- docker command, 694, 697



docker daemon, 694  
 Docker Desktop, 768  
 Docker Hub, 696  
 Docker project, 694  
 drivers, printer drivers, 404  
 du command, 334–335  
 dual booting, 208–209  
 dumb terminals, 137  
 dump/restore, 566  
 DVD, installing Linux, 196  
 DVD drive, 197

## E

echo statement, 148  
 ecryptfs, 613–615  
 emacs editor, 114  
 email, encrypting messages, 607–608  
 encryption/decryption  
   cipher keys  
     asymmetric keys, 604–605  
     email message encryption, 607–608  
     key pair generation, 605–606  
     public key sharing, 607  
     symmetric keys, 603–604  
     tar archive files, 604  
   ciphers, 602–603  
   digital signatures, 608–610  
 enterprise  
   cloud-init, 738  
   containers and, 706  
   installing Linux, 196  
   network configuration  
     Linux as DHCP server, 365  
     Linux as DNS server, 365–366  
     Linux as proxy server, 366  
     Linux as router, 364  
   Samba and, 497  
   server management, 336  
   software management, 245–246  
   user management, 261  
     ACLs (Access Control Lists), 262–269  
     permission setting, 262–269  
 enterprise networking, 340  
 environment variables, 81, 82–83  
   PATH, 70  
   shell, adding, 87  
 escaping shell characters, 149  
 /etc directory, 94  
   aliases file, 181  
   bashrc file, 181  
   crontab file, 182  
   csh.cshrc file, 182  
   exports file, 182  
   fstab file, 182  
   group file, 182  
   gshadow file, 182  
   host.conf file, 182  
   hostname file, 182  
   hosts file, 182  
   inittab file, 182  
   mtab file, 182  
   mttools.conf file, 182  
   named.conf file, 182  
   nsswitch.conf file, 182  
   ntp.conf file, 182  
   passwd file, 182  
   printcap file, 183  
   profile file, 183  
   protocols file, 183  
   rpc file, 183  
   rsyslog.conf file, 183  
   services file, 183  
   shadow file, 183  
   shells file, 183  
   sudoers file, 183  
   xinetd.conf file, 183  
 /etc/cron directory, 180  
 /etc/cups directory, 180, 405  
 /etc/default directory, 180  
 /etc/exports file  
   configuring, 504  
   hostnames, 505–506  
   nfsnobody, 506  
   options, 506  
   root user, 506  
   user mapping options, 506  
 /etc/fstab file, mountable file systems, 295–297  
 /etc/hostname file, 358  
 /etc/hosts file, 358  
 /etc/httpd directory, 181  
 /etc/mail directory, 181  
 /etc/nsswitch.conf file, 359–360  
 /etc/postfix directory, 181  
 /etc/ppp directory, 181  
 /etc/rc?.d directory, 181  
 /etc/rc.d/init.d directory, 375–376  
 /etc/resolv.conf file, 359

- `/etc/samba/smb.conf` file
  - [global], 486–487
  - [homes], 486, 487–489
  - [printers], 486, 489–493
- `/etc/security` directory, 181
- `/etc/services` file, 663–664
- `/etc/skel` directory, 181
- `/etc/sysconfig` directory, 181
- `/etc/sysconfig/network` file, 358
- `/etc/systemd` directory, 181
- `/etc/X11` directory, 183
- `/etc/xinetd.d` directory, 181
- eth0 interface, 362
- Ethernet, 339
  - channel bonding, 361–362
- Execute permissions, 106
- exercise answers
  - Ansible, 858–860
  - application automation, 858–860
  - cloud computing, 855–856
    - deploying to cloud, 857
    - shifting to, 853–854
  - desktop creation, 797–800
  - disk management, 819–821
  - file management, 819–821
  - filesystem, 802–803
  - FTP server configuration,
    - 835–838
  - infrastructure automation, 858–860
  - Kubernetes, 860–861
  - Linux installation, 812–813
  - network administration, 825–827
  - NFS file server configuration, 841–843
  - print server configuration, 829–831
  - processes, running, 805–807
  - Samba server configuration, 838–841
  - security
    - advanced, 847–849
    - basic, 845–847
    - network security, 851–853
    - SELinux and, 849–851
  - server administration, 822–825
  - services, starting/stopping, 827–829
  - shell, 800–802
  - shell script writing, 807–810
  - software acquisition, 814–815
  - software management, 814–815
  - system administration, 810–812
  - text files, 804–805
  - troubleshooting, 843–845

- user account management, 815–819
  - web server configuration, 831–835
- exiting shell, 83–84
- expanding commands, 78, 80
- expanding parameters, 151–152
- expanding variables, 80–81
- exportfs command, 507
- exporting shared filesystems, 507
- exports file, 182
- exports man page, 504
- expressions
  - arithmetic, expanding, 80
  - test expressions, 154
  - operators, 155–156
- extended memory, 4

## F

- FCoE (Fibre Channel over Ethernet Devices), 213
- Fedora, 18–19, 27
  - downloading, 788–789
  - installing from Live media
    - bare metal system, 198
    - multi-boot, 198
    - single-boot, 198
    - virtual system, 198
  - Terminal window, 64
- Fibre Channel, 5
- file-matching metacharacters, 98–99
- filenames, shell scripts, 148
- file-redirection metacharacters, 99–100
- files
  - commands, 96–98
  - copying, 110
    - scp command, 321–324
  - encrypting, 616
  - listing, 101–105
  - moving, 109–110
  - Nautilus, 42, 43
    - filesystem organization, 42–43
  - ownership
    - Apache web server, 433
    - changing, 109
  - password files, 574–576
  - permissions, 105–106
    - Apache web server, 433
    - changing, chmod, 106–108
    - default, 108–109
    - errors, 452
    - Execute, 106

- Read, 106
  - vsftpd, 465
  - Write, 106
  - removing, 110
  - searching for
    - by date and time, 125–126
    - find command, 122–128
    - grep command, 128–129
    - locate command, 120
    - by name, 123–124
    - not, 126–127
    - or, 126–127
    - by permission, 125
    - by size, 124
    - by user, 124
  - filesystems, 4, 93, 273, 275, 500
    - commands, 71, 96–98
    - creating, 300
    - directories, 94
    - encryption at installation, 611–613
    - Linux compared to Windows-based, 95
    - mounting, 291–293
      - autoofs, on demand, 517–520
      - defining mountable systems, 295–297
      - /etc/fstab file, 295–297
      - mount command, 297–298
      - NFS, 512–520
      - options, 515–517
      - swap areas
      - disabling, 294–295
      - enabling, 293–294
    - Nautilus organization, 42–43
    - noauto, 514–515
    - partitions, 273
    - root directory, 93
    - security
      - dangerous permissions, 576–577
      - lockdown, 578–579
      - securing files, 577–578
    - shared, exporting, 507
    - system administrator and, 168
  - filter table, 678
  - find command, 122–128, 335–336
  - Firefox, FTP access, 470
  - firefox package, 227
  - Firewall Configuration window, 509, 674–675
  - firewall-config command, 674–675
  - firewalld service, 674–675, 675–677
  - firewalls, 313, 672–674
    - Apache web server, 433–434
    - application-layer firewalls, 674
    - implementing, 674–688
    - iptables, 673
    - iptables, 313
    - network-layer firewalls, 674
    - rules, Cockpit and, 677–678
    - Samba, 482–483
    - troubleshooting and, 552–553
  - firmware
    - RAID devices, 213
    - starting from, 526–528
  - folders, Nautilus, 42, 43
    - creating, 43
    - Home folder, 42
  - for...do loop, 157–158
  - foreground processes, 137–138
    - commands, 139–140
  - FOSS (Free and Open Source Software), 12, 666
  - free distribution, GPL and, 12
  - free software, 12
  - Free Software Directory, 12
  - FreeBSD, 13
  - FSF (Free Software Foundation), 11
  - fstab file, 182
  - FTP (File Transfer Protocol), 455–456
    - active connection, 456
    - clients, 469–473
    - command-oriented clients, 456
    - containers, GitHub and, 703–705
    - passive connection, 456
    - server, 309
      - accessing, 470–472
      - anonymous, 456
      - firewall, 461–463
      - gFTP client, 472–473
      - graphical tools, 456
      - SELinux, configuring, 463–465
      - uploading, allowing, 467–468
      - user access, setup, 465–466
      - vsftpd, 457–461
    - servers, running from container, 699–701
  - functions, 71
    - completion, 75
- ## G
- gconf-editor, 173
  - gedit command, 113–114
  - general regular expression print. *See* grep command

- Gentoo, 16, 207
  - gFTP client, 472–473
  - Gibson Research Corporation, 596
  - GitHub, FTP containers, 703–705
  - GlusterFS, 5
  - GNOME, 30–31
  - GNOME 2
    - Appearances Preference, 49
    - Compiz, 46
    - GNOME panels, 47
    - Metacity, 46, 48–49
    - Nautilus, 46
    - panels, 50
      - adding, 52
      - applets, 51
      - application launcher, 52–53
      - Applications menu, 51
      - drawers, 53
      - moving items, 50
      - properties, 54
      - resizing items, 51
      - System menu, 51
      - Window list, 51
    - Preferences, 47
  - GNOME 3, 31
    - applications, 41
      - additional, 34
      - launching, 37–38
      - opening, 32
    - Applications view, 37
    - Bluetooth, 39
    - bootup, 31
    - commands, launching, 37–38
    - dash, 36
    - devices, 39
    - keyboard, Windows key, 36
    - Nautilus
      - files, 42–43
      - folders, 42, 43
      - FTP with login, 43
      - Public FTP, 43
      - remote content, 43
      - Rhythmbox, 45–46
      - Secure (HTTPS), 43
      - software, 43–45
      - SSH and, 43
      - WebDav (HTTP), 43
      - Windows share, 43
    - navigation
      - keyboard, 36–38
      - mouse, 32–35
      - networking, 39
      - searches, 37
      - set up, 38–39
      - shell extensions, 39–40
      - sound, 39
      - stopping, 46
      - System Settings window, 38
      - toggling, 32
      - top bar, 36
      - Tweak Tool, 40–41
      - views, 36
      - window menu, 35
      - windows
        - active, 37
        - minimized, 33
        - opening, 32
        - Windows view, 36
        - workspaces, multiple, 34–35
  - GNOME Terminal, 64
  - gnome-disks, 173
  - gnome-utils, 173
  - GNU (GNU is Not UNIX), 11–12
    - BSD (Berkeley Software Distribution) License, 15
    - LGPL (Lesser General Public License), 15
    - MIT license, 15
    - Mozilla license, 15–16
  - GNU Project page, 11
  - GPL (GNU Public License), 12
  - graphical tools, 172
  - graphical windows, 168
  - graphics, Red Hat, 17
  - greater than (>), 78
  - grep command, 128–129, 159
  - group accounts, 259–261
  - group file, 182
  - groupadd command, 260–261
  - groups, 249
  - GRUB (GRand Unified Bootloader), 217–218, 528–530
    - troubleshooting, 528–530
  - GRUB 2 boot loader, 530–531
    - troubleshooting, 530–531
  - gshadow file, 182
  - GUID (Globally Unique Identifier), partition tables, 276
  - GUIs (graphical user interfaces), 61
- ## H
- hard drive, partitioning
    - assigning to directory, 216
    - filesystem types, 214
    - Linux partitions, 215

- LVM partitions, 215
  - multiple operating systems, 214
  - multiple-partition disks, 281–285
  - partition tables, 275–276
  - RAID partitions, 215
  - single-partition disks, 277–281
  - swap partitions, 215
  - viewing partitions, 276–277
  - hardware, 4
    - checking, 187–189
    - kernel and, 186–193
    - modules, loadable, 191–193
    - removable, 189–191
    - requirements, 196–197
  - hashed passwords, 574–576
  - headers, packet headers, 673
  - help command, 88
  - here text, 100
  - hierarchy of directories, 94
  - history command, 72–78
  - /home directory, 94
  - host systems, generic, 336
  - host.conf file, 182
  - hostname
    - completion, 75
    - /etc/exports, 505
  - hostname file, 182
  - hosts
    - individual, 505
    - IP network, 505
    - TCP/IP domain, 505
  - hosts file, 182
  - httpd daemon, 442–443
  - httpd package, 310
  - hybrid cloud, 731
  - hypervisor, 709, 713
    - configuring, 715–718
    - DNS, setup, 718
    - /etc/hosts, editing, 718
    - Linux, installing, 716–717
    - naming, 717
    - services, 717–718
    - setup, 714
- I**
- id command, 69
  - IDS (Intrusion Detection System), 592–595
  - if...then statements, 153–154
  - images, 694
    - building, 702–703
    - cloud
      - Amazon EC2, 744–746
      - OpenStack and, 739–744
    - for clouds, 731–733
    - container registries and, 705–706
    - virtual machines, 721
  - implicit classes, 404
  - Infiniband, 5
  - info command, 89
  - information about commands, 88–90
  - init, 369, 370–371, 371–377, 524–525
    - runlevels, 373–374
    - systemd, 370–371
    - SysVinit, 370
    - troubleshooting, 533
  - inittab file, 182
  - input/output redirection, 8
  - installation
    - Apache HTTPD web server, 431
    - boot options
      - feature disable, 210
      - kickstarts, 211–212
      - mediacheck, 212
      - rescue mode, 212
      - special installation, 210–211
      - video problems, 210
    - dual booting
      - defragmenting, 209
      - hard disk, adding, 208
      - Windows partition resize, 208
    - GRUB (GRand Unified Bootloader), 217–218
    - NFS server, 502
    - nmap utility, 665–666
    - Samba, 476–478
    - from scratch, 207
    - servers, 308–310
    - software, 221–222
    - storage, specialized, 213
  - installation server, 206
  - installing Linux
    - cloud based installations, 204–205
    - from DVD, 196
      - Red Hat Enterprise, 201–204
    - in enterprise, 196, 205–207
    - GRUB (GRand Unified Bootloader), 217–218
    - hard drives, partitioning, 214–217
    - from Live media, 195
      - Fedora installation, 198–201
    - from scratch, 207
    - virtualization and, 209
  - integer arithmetic, 152–153

- interactive copying, 324
- interfaces, 4
- interpreter, shell script, 148
- IP (Internet Protocol), 341
  - addresses
    - aliases, setting, 350–351
    - manually setting, 349–350
    - routes, setting, 351–352
    - source, blocking, 684–685
- IP masquerading, 673
- IPC (interprocess communications), 695
- IPP (Internet Printing Protocol), 404
- iptables firewall, 313
- iptables utility, 673, 674, 680
  - chains, 679
  - configuration, saving, 687–688
  - DROP, 683
  - filter table, 678
  - mangle table, 678
  - nat table, 678
  - options, 683
  - policies, modifying, 680–683
  - port blocking, 685–687
  - protocol blocking, 685–687
  - raw table, 678
  - rules, modifying, 680–683
  - security table, 678
  - source IP address blocking, 684–685
- iSCSI, 5, 213
- ISO images, 787
- J**
  - Java, JBoss, 18
  - JBoss, 18
  - journalctl command, 184
- K**
  - Kali Linux, 595
  - KDE (K Desktop Environment), 29
  - Kerberos, 309, 712
  - kernel, 13, 16
    - hardware and, 186–193
    - ring buffer, 532
    - starting, 532–541
    - startup, 532–541
  - Kernighan, Brian, 9
  - key-based authentication, Secure Shell tools, 324–326
  - keystrokes
    - command history, 77–78
    - command-line editing, 74
  - kickstart files, 206
    - boot options and, 211–212
    - Linux installation, 205
    - RPM packages and, 246
  - killall command, 141–142
  - killing processes
    - kill command, 140
    - killall command, 140, 141–142
  - KNOPPIX, 16
  - ksh (Korn shell), 61, 65
  - Kubernetes, 5, 765–766
    - accessing, 769–771
    - applications, 767–768
    - clusters, 766
    - container engines, 767
    - containers and, 694
    - Docker Desktop, 768
    - interfaces, 768
    - Minikube and, 766, 768
      - starting, 770–771
    - nodes
      - master node, 766–767
      - worker node, 766, 767
    - OpenShift and, 782–783
    - Pods, 766
    - services, 766
    - sidecar containers, 766
    - storage, 766
    - tutorials, 768
  - Kubernetes Basics Tutorial, 769–770, 771–772
    - application deployment, 772–773
    - applications
      - exposing, 776–777
      - scaling down, 781–782
      - scaling up, 779–780
    - load balancer, 780–781
    - pod information, 773–776
    - services
      - deleting, 778–779
      - labeling, 777–778
  - KVM (Kernel-based Virtual Machine), 5, 209, 710, 713
- L**
  - LAMP (Linux, Apache web server, MySQL database, PHP web scripting language) stack, 3
  - laptops, networking, 340
  - LDAP (Lightweight Directory Access Protocol), 270, 309

LDP/LPR printers, 413  
 left parenthesis ( ( ), 78  
 less than (<), 78  
 lftp command, 470–472  
 /lib directory, 94  
 Libvirt Service Daemon, 713  
 libvirtd service, 713, 715–718  
 libvirt-daemon-config-network package,  
     717  
 Linux  
     booting from USB drive, 791–792  
     compared to other OSs, 6  
     features, 4–5  
     history, 3  
         Bell labs, 7–8  
         BSD, 12–13  
         commercial UNIX, 9–11  
         GNU, 11–12  
         OSI, 14–16  
         UNIX, 7–8  
     kernel, 13  
     Minix, 7  
     as open source UNIX-like OS, 14  
 Linux Foundation, 14  
 Linux partitions, 215  
 listing  
     directories, 101–105  
     files, 101–105  
 Live media, installing Linux from, 195  
     Fedora, 198–201  
 locate command, 72, 120–122  
 logwatch service, 331–332  
 loopback, mounting disk image, 298–299  
 loops  
     for...do, 157–158  
     until...do, 158–159  
     while...do, 158–159  
 lp command, 419  
 lprm command, 419–420  
 lpstat command, 419  
 ls command, 67–68, 101–105  
 LUKS (Linux Unified Key Setup), 612  
 LVM (Logical Volume Manager), 273, 274,  
     538  
     partitions, 215, 285  
         creating logical volumes, 289–290  
         displaying, 286–288  
         volume growth, 290–291  
     physical volumes, 273  
 LXDE (Lightweight X11 Desktop Environment), 29

## M

MAC addresses, 341  
 mail server, 309  
 man command, 89  
 man pages, 502  
     exports, 504  
     sections, 89  
 Mandrake, 16  
 Mandriva, 16  
 mangle table, 678  
 mangling packet headers, 673  
 master nodes, 336  
 MBR (Master Boot Record), 275  
 MCS (Multi-Category Security), 638  
 /media directory, 94  
 mediacheck, 212  
 memory, 4  
     OOM condition, 556  
     page caches, 558  
     processes, killing, 558  
     troubleshooting, 553–559  
 metacharacters, 78  
     brace expansion, 101  
     file-matching, 98–99  
     file-redirection, 99–100  
 meta-data, 730  
 Microsoft Active Directory. *See* AD (Active Directory)  
 migration, VMs, 725–727  
 Minikube, 766, 768  
     starting, 770–771  
 Minix, 7  
     /misc directory, 94  
 mkfs command, 300  
 MLS (multi-level security), 638–639  
     /mnt directory, 94  
 modules  
     loaded, listing, 191–192  
     loading, 192  
     removing, 192–193  
 monitoring servers  
     Cockpit, 314  
     crackers, 315  
     logging configuration, 314  
     software updates, 315  
     system activity reports, 314  
 mount command, 297–298  
 mounting, 274  
     disk images, in loopback, 298–299  
     filesystems, 291–293

- mounting (*continued*)
    - defining mountable systems, 295–297
    - /etc/fstab file, 295–297
    - mount command, 297–298
    - options, 515–517
    - swap areas, 293–295
    - unmounting NFS, 520–521
  - NFS
    - autofs, on demand mounting, 517–520
    - at boot time, 513–517
    - manually, 512–513
    - umount command, 299
  - mount-level security, 499
  - moving files, 109–110
  - MS-DOS filesystems, 95
  - MTA (Mail Transport Agent) server, 309
  - mtab file, 182
  - mtools.conf file, 182
  - Multics, 8
  - multipath devices, 213
  - multitasking, 167
  - multiuser features, 167
  - mv command, 109–110
- ## N
- named.conf file, 182
  - namespaces, containers, 694, 695
  - nano editor, 114
  - nat table, 678
  - Nautilus
    - files, 42, 43
    - filesystem organization, 42–43
    - folders, 42, 43
      - creating, 43
      - Home folder, 42
    - FTP with login, 43
    - Public FTP, 43
    - remote content, 43
    - Rhythmbox, 45–46
    - Secure (HTTPS), 43
    - software
      - installing, 43–45
      - managing, 43–45
    - SSH and, 43
    - WebDav (HTTP), 43
    - Windows share, 43
  - NCSA (National Center for Supercomputing Applications), 428
  - NetBEUI, 475
  - NetBSD, 13
  - netfilter/iptables tables, 678–679
    - policies, 679–680
    - rules, 679–680
    - targets, 679–680
  - network bridge, VMs, 721
  - network cards, 197
  - network interfaces
    - aliases, 360–361
    - bonded, 362
    - Cockpit and, 343–345
    - command line and, 345–349
    - configuring
      - IP address aliases, 350–351
      - IP address manual set, 349–350
      - route setting, 351–352
    - domain names, 349
    - host names, 349
    - NetworkManager and, 342–343
    - routing information, 347–348
    - troubleshooting and, 547–548
    - viewing, from command line, 345–347
  - Network Mapper, 665
  - network services, auditing, 663–665
    - nmap utility, 665–672
  - networking
    - cloud, setup, 714
    - configuration
      - command line, 353–364
      - enterprise, 364–366
      - files, 355–360
    - custom routes, 363–364
    - desktop, 340–353
    - enterprise, 340
    - Ethernet channel bonding, 361–362
    - hostnames, troubleshooting and, 549–550
    - laptop, 340
    - physical connections, troubleshooting and, 548
    - proxy connections, configuring, 352–353
    - routes
      - custom, 363–364
      - troubleshooting and, 548–549
    - servers, 340
    - troubleshooting
      - incoming connections, 550–553
      - outgoing connections, 547–550
  - network-layer firewalls, 674
  - NetworkManager, 340



- DHCP
  - server response, 340
  - service request, 340
- domain name server, 341
- gateway, default, 341
- IP address, 341
- lease time, 341
- local settings, 342
- network interfaces, activating, 340
- network settings, 350
- subnet mask, 341
- NetworkManager TUI, connection, editing, 354–355
- NFS (Network File System) server, 309, 499
  - filesystems, sharing, 503–507
  - installing, 502
  - mounting
    - autofs, on demand mounting, 517–520
    - at boot time, 513–517
    - client, 500
    - manually, 512–513
  - security, 508
    - filesystem structure exposure, 508
    - firewall, 508–510
    - root users, 508
    - SELinux configuration, 511–512
    - TCP wrappers, 510–511
    - unencrypted communications, 508
    - user mapping, 508
  - shares, viewing, 512
  - unmounting, 520–521
- nfs-server service, starting, 502–503
- nfs-utils package, 502
- nice command, 142–143
- NIS (Network Information Service), 270
  - groups, 506
- nmap utility
  - installing, 665–666
  - port scans, 666
  - port states, 667
- nmbd service
  - starting, 480–481
  - stopping, 481–482
- nmtui command, 354
- noauto filesystem, 514–515
- Nokia, 9
- nsswitch.conf file, 182
- ntp.conf file, 182
- ntpd package, 309

## O

- OEM (original equipment manufacture), 10
- one-command actions, 156
- OOM condition, 556
- open source, storage platforms
  - Ceph, 5
  - GlusterFS, 5
- Open Source Development Labs, 14
- open source software, 12
- OpenBSD, 13
- OpenPGP, 616
- OpenShift, 173
  - Kubernetes and, 782–783
- openssh package, 316
- openssh-clients package, 316
- openssh-server package, 316–318
- OpenStack, 5
  - cloud images, 739–744
  - remote access keys, 741–742
  - VM access via ssh, 743–744
  - VM launch, 742–743
- operators
  - Ansible, 749–750
  - expressions, test expressions, 155–156
- /opt directory, 94
- options, commands, 67–68
- OSI (Open Source Initiative), 14–16
- OSs (operating systems), Linux and, 6
- oVirt project, 5
- ownership of files, 109

## P

- package collections, 308
- packages
  - nfs-utils, 502
  - openssh, 316
  - openssh-clients, 316
  - openssh-server, 316
- servers
  - directory server, 309
  - DNS server, 309
  - FTP server, 309
  - mail server, 309
  - Network Time Protocol server, 309
  - NFS file server, 309
  - print server, 309
  - rsyslog service, 308–309

- packages (*continued*)
  - SQL server, 309
  - system logging server, 308–309
  - web server, 309
  - Windows file server, 309
- packet filters, 674
- packet headers, mangling, 673
- packets, blocking/allowing, 673
- PAM (Pluggable Authentication Module), 312
  - administering
    - application configuring files, 622–623
    - password enforcement, 628–632
    - sudo and, 632–633
    - system event configuration files, 623–626
    - time restrictions, 626–627
  - authentication process, 619
    - contexts, 619–620
    - control flags, 620–621
    - modules, 621–622
  - resources, 633
- PAM facility, 312
- panels (GNOME), 50
  - adding, 52
  - applets, 51
  - application launcher, 52–53
  - Applications menu, 51
  - drawers, 53
  - moving items, 50
  - properties, 54
  - resizing items, 51
  - System menu, 51
  - Window list, 51
- parameters, shell script
  - expanding, 151–152
  - reading in, 151
- partition tables, GUID, 276
- partitioning
  - filesystems, 273
  - hard drives
    - assigning to directory, 216
    - filesystem types, 214
    - Linux partitions, 215
    - LVM partitions, 215
    - multiple operating systems, 214
    - RAID partitions, 215
    - swap partitions, 215
  - hypervisors and, 717
  - LVM (Logical Volume Manager), 285
    - creating volumes, 289–290
    - displaying, 286–288
    - volume growth, 290–291
  - multiple-partition disks, 281–285
  - partition tables, 275–276
  - single-partition disks, 277–281
  - viewing, 276–277
- passwd file, 182
- passwords, 312–313
  - changing, 571–572
  - enforcing best practices, 572–574
  - files, 574–576
  - hashes, 574–576
  - PAM, 628–632
  - public key authentication, 313
  - selecting, 570–571
  - setting, 571–572
- path, 70
  - absolute, 96
  - order, 71
- PATH environment variable, 70, 88
- PE (Physical Extent), 286
- penetration testing, 595
- permissions, 105–106
  - changing, `chmod`, 106–108
  - daemons and, 311
  - default, `umask` value, 108–109
  - errors, 452
  - Execute, 106
  - Read, 106
  - Write, 106
- persistent services, enabling, 391–394
- physical security, 565–566
- PID (process ID), 131, 370
- pipe character ( `|` ), 78
- plain-text files, 179
- playbooks (Ansible), 749
  - creating, 757–758
  - imports, 753
  - includes, 753
  - modules, 752–753
  - plays, 752
  - roles, 753
  - running, 758–760
  - tasks, 752
- podman command, 694, 697
- port numbers, daemon processes, 311
- portability, 8–9
- ports, 666

- auditing, 666
  - blocking, 685–687
  - scans
    - TCP Connect, 666
    - UDP, 666
  - states, 667
  - positional parameters, 150–151
  - POSIX (Portable Operating System Interface), 10
  - PostgreSQL, 309
  - print server, 309
    - configuring
      - shared CUPS printer, 420–422
      - shared Samba printer, 422–424
  - Print Settings window, 403
    - local printers
      - adding, 409–411
      - editing, 411–412
    - remote CUPS printers, 413
    - remote LDP/LPR printers, 413
    - remote printers, configuring, 412–413
    - remote UNIX printers, 413
    - Windows (SMB) printer, 414–415
  - printcap file, 183
  - printer browsing, 404
  - printer classes, 404
  - printers, adding automatically, 405–406
  - printing. *See also* CUPS (Common UNIX Printing System)
    - canceling jobs, 408
    - commands, 418
      - lp, 419
      - lprm, 419–420
      - lpstat, 419
    - drivers, 404
    - /etc/cups directory, 405
    - IPP (Internet Printing Protocol), 404
    - listing print jobs, 407
    - moving jobs, 408
    - printer classes, 408
    - UNIX print commands, 404
    - viewing printers, 408
  - private cloud, 730
  - private key cryptography, 603
  - /proc directory, 95
  - processes, 4, 131–132
    - background, 137–138
      - commands, 139–140
    - bigcommand, 141
    - cgroups, 143–144
    - changing, 134–135
    - daemon processes, 5, 179
    - foreground, 137–138
      - commands, 139–140
    - killall command, 141–142
    - killing, 135, 558
      - kill command, 140
      - killall command, 140
    - limiting, 143–144
    - listing
      - ps command, 132–134
      - System Monitor, 136–137
      - top command, 134–135
    - ( | ) piping, 133
    - priorities, 142–143
    - renicing, 135
    - RSS (resident set size), 133
    - VSZ (virtual set size), 133
  - processors, 196
  - professional opportunities, 19–21
  - profile file, 183
  - programming, utilities, 5
  - prompt, setting, 85–87
  - protocols, blocking, 685–687
  - protocols file, 183
  - proxy servers, configuring connections, 352–353
  - ps command, 132–134
  - public cloud, 730
  - public key authentication, 313
  - pulling containers, 697–698
  - pwd command, 67
  - PXE server, 206
    - RPM packages and, 246
- ## Q
- QEMU, 713
  - qemu process, 713
  - question mark (?), 99
  - Qwest, 9
- ## R
- RAID partitions, 215
  - RAM (random access memory), 197, 273–274
    - troubleshooting and, 554–555
  - raw table, 678
  - RBAC (role-based access control), 635
    - TE (type enforcement) and, 637–638
  - rc.sysinit, troubleshooting, 533–534

- Read permissions, 106
- real-time computing, 5
- Red Hat
  - Anaconda installer, 17
  - graphical administration, 17
  - RPM package management, 16–17
- Red Hat Enterprise
  - downloading, 789–790
  - installing, from DVD, 201–204
- Red Hat OpenShift, 18
- Red Hat OpenStack Platform, 18
- Red Hat Virtualization, 5
- remote access, 307. *See also* Secure Shell tools
  - interactive copying, 324
- removable hardware, 189–191
- Removable Media window, 189–190
- renice command, 142–143
- repositories, software, 223
- rescue mode, 212
  - troubleshooting in, 559–561
- reserved words, 71
- REST API, Ansible Tower and, 763
- restricted deletion directory, 268–269
- reviews
  - compliance, 595–596
  - security, 596
- RHCE (Red Hat Certified Engineer), 21, 22
  - network services, 24–25
  - system configuration and management, 24
  - topics, 23–25
- RHCSA (Red Hat Certified System Administrator), 21, 22
  - topics, 22–23
- RHEL (Red Hat Enterprise Linux), 17–18
  - Terminal window, 64
- RHELOSP (Red Hat Enterprise Linux OpenStack Platform), 173
- RHEV (Red Hat Virtualization), 173
- Rhythmbox (Nautilus), 45–46
- right parenthesis ( ) , 78
- Ritchie, Dennis, 8, 9
- rlogin, 316
- rm command, 110
- root directory, 93
- /root directory, 95
- root user, 174
  - sudo facility, 176–178
  - via GUI, 176
  - via shell, 175–176
- rootkits, 590–595

- route-interface file, 363
- rpc file, 183
- rpcbind service, 503
- rpm command
  - package installation, 241–242
  - package removal, 241–242
  - querying information, 242–244
- RPM package management, 16–17
- RPM (RPM Package Manager) packaging, 225, 226–228
  - container images, 246
  - dependencies, 228
  - installing, 228
  - kickstart files, 246
  - location, 228
  - origins, 227–228
  - PXE boot, 246
  - satellite server, 246
  - Spacewalk, 246
  - verifying packages, 244–245
  - YUM and, 229–232
    - third-party software repositories, 233
  - transition to DNF, 229
  - yum command, 233–241
- RSS (resident set size), 133
- rsync command, 322–323, 566
- rsyslog service, 308, 326–331
- rsyslog.conf file, 183, 327–329
- rsyslogd daemon, 326–327
- rsyslogd facility and, 184–185
- runlevels, 369, 373–374
  - default, 369
  - configuring, 394–395
  - troubleshooting, 534–538

## S

- Samba, 475–476
  - enterprise, 497
  - /etc/samba/smb.conf file
    - [global], 486–487
    - [homes], 486, 487–489
    - [printers], 486, 489–493
  - firewalls, configuring, 482–483
  - folders
    - checking share, 490–493
    - shared, 489–490
  - host/user permissions, 486
  - installing, 476–478
  - printer configuration, 422–424

- SELinux, configuring, 484–486
- share access
  - Linux file manager, 493–495
  - mounting from command line, 495–496
  - Windows, 496–497
- starting, 478–480
- stopping, 481–482
- samba package, 309, 476–478
- samba-client package, 476
- samba-common packages, 476
- samba-winbind package, 476
- SAN devices, 213
- SANS institute, 596
- sar command, 332–333
- satellite server, RPM packages and, 246
- /sbin directory, 95
- SCO (Santa Cruz Operation), 10
- scp command, file copy and, 321–324
- scripts, backup script, 162
- secon command, 648–649
- secret key cryptography, 603
- Secure Shell tools, 316
  - client tools, 318–324
  - key-based authentication, 324–326
- security, 307
  - Apache web server
    - file ownership, 433
    - file permissions, 433
    - firewalls, 433–434
    - SELinux and, 434–435
  - Bell-LaPadula Mandatory Access model, 639
  - cryptography, 599–600
    - block ciphers, 600
    - ciphers, 600
    - decryption, 600
    - encryption/decryption, 602–610
    - hashing, 600–601
    - implementing, 610–618
    - stream ciphers, 600
  - disaster recovery, 566
  - filesystem, 576–579
  - firewalls, 672–674
    - application-layer firewalls, 674
    - Cockpit and, 677–678
    - implementing, 674–688
    - iptables, 673
    - network-layer firewalls, 674
  - MCS (Multi-Category Security), 638
  - MLS (multi-level security), 638–639
  - mount-level, 499
  - network services, auditing, 663–665, 665–672
  - PAM (Pluggable Authentication Modules), 618
    - administering, 622–633
    - authentication process, 619–622
    - resources, 633
  - passwords, 570–576
  - physical, 565–566
  - servers, 312–314
    - configuration file settings, 314
    - firewalls, 313
    - passwords, 312–313
    - SELinux, 313
    - TCP Wrappers, 313
  - services, 579–580
  - software, 579–580
  - system administration and, 169
  - systems monitoring, 580–581
    - filesystem, 587–595
    - log files, 581–584
    - user accounts, 584–587
  - user accounts, 566–569
- security clearance, 639
- security reviews, 596
- security table, 678
- sed command, 160
- SELinux (Security Enhanced Linux), 313
  - benefits of, 635–636
  - Booleans and, 653–654
  - Booleans for Samba, 484–485
  - configuration, NFS server, 511–512
  - errors, 452
  - file contexts for Samba, 485–486
  - least privilege access, 636
  - mode, setting, 645–647
  - monitoring, 654–656
  - operational modes
    - disabled mode, 639–640
    - enforcing mode, 640
    - permissive mode, 640
  - policy rules, 644–645
    - package management, 651–653
  - policy types, 643
    - minimum, 644
    - MLS (Multi-Level Security), 644
    - setting, 647–648
    - targeted, 644
  - process sandboxing, 636
  - RBAC, 636

- packages (*continued*)
  - resources, 659
  - security contexts, 640–641
    - file security, 650–651
    - files, 642
    - level attribute, 641
    - processes, 642–643
    - role attribute, 641
    - secon command, 648–649
    - type attribute, 641
    - user, 649–650
    - user attribute, 641
    - users, 641–642
  - security models, implementing, 639–645
  - TE (type enforcement), 637–638
  - testing, 636
  - troubleshooting
    - Booleans set incorrectly, 658–659
    - context labels, lost, 658
    - logging, 656–657
    - nonstandard directory as service, 657
    - nonstandard port as service, 658
- semicolon character (;), 78, 79
- sequential commands, 79
- servers, 307
  - checking, 316
  - configuring, 310–311
  - enterprise and, 336
  - installation server, 206
  - installing, 308–310
  - monitoring, 314–315
    - Cockpit, 314
    - crackers, 315
    - logging configuration, 314
    - software updates, 315
    - system activity reports, 314
  - network, 340
- packages
  - directory server, 309
  - DNS server, 309
  - FTP server, 309
  - mail server, 309
  - Network Time Protocol server, 309
  - NFS file server, 309
  - print server, 309
  - rsyslog service, 308–309
  - SQL server, 309
  - system logging server, 308–309
  - web server, 309
  - Windows file server, 309
- PXE, 206
- securing, 312–314
  - configuration file settings, 314
  - firewalls, 313
  - passwords, 312–313
  - SELinux, 313
  - TCP Wrappers, 313
- setting, 316
- starting, 311–312
- system administration and, 169
- services
  - firewalld, 674–675, 675–677
  - persistent, enabling, 391–394
  - reloading, 388
  - security, 579–580
  - starting, 5
  - status, checking, 384–387
  - systemd, new, 399–401
  - SysVinit
    - new, 396–398
    - starting and stopping, 387–391
- services file, 183
- set GID bit, 267–268
- set UID bit, 268
- setfacl command, 262–264
- sftp command, 324
- sh shell, 65
- shadow file, 183
- shared filesystems, exporting, 507
- shell, 4, 61
  - # prompt, 63
  - \$ prompt, 63
  - accessing, 62–63
  - ash, 61
  - bash shell, 61
    - configuration files, 84
  - Bourne shell, 61
  - command languages and, 62
  - configuring, 84–85
  - csh (C shell), 61, 65
  - dash, 65
  - dash shell, 61
  - environment, 84–87
  - environment variables, 87
  - exiting, 83–84
  - ksh, 65
  - metacharacters

- brace expansion, 101
- file-matching, 98–99
- file-redirection, 99–100
- prompt, 63
  - setting, 85–87
- reasons to learn, 62
- root user, 175–176
- running from container, 698–699
- selecting, 65–66
- sh, 65
- tcsh, 61, 65
- variables, 81
- shell history, 72–73
- shell script
  - arithmetic in, 152–153
  - backup script, 162
  - case command, 156–157
  - chmod command, 162
  - command-line argument, 148, 150–151
  - cut command, 159
  - for...do loop, 157–158
  - grep, 159
  - if...then statements, 153–154
  - interpreter, 148
  - parameters
    - expanding, 151–152
    - reading in, 151
  - positional parameters, 150–151
  - programming constructs, 153–159
  - sed command, 160
  - special characters, escaping, 149
  - telephone list example, 161–162
  - text
    - cutting, 159–160
    - deleting, 160
    - translating, 160
  - text manipulation, 159–161
  - tr command, 160
  - until...do loop, 158–159
  - variables, 149
  - while...do loop, 158–159
- shell scripts, 147–148
- shells file, 183
- sidecar containers, 766
- signals, 141
- single-partition disks, 277–281
- Slackware, 16
- SLES (SUSE Linux Enterprise Server), 730
- SMB (Server Message Block), 475
- smb.conf file, 422–423
- software
  - bounties, 21
  - dependent software, 224
  - free software, 12
  - installation, 221–222
    - system administrator and, 168
  - Nautilus, 43–45
  - open source software, 12
  - repositories, 223
  - security, 579–580
    - advisories, 580
    - package updates, 579–580
  - subscriptions, 20–21
  - upstream software providers, 228
- Software window, 222–223
- source code, UNIX and, 10
- Spacewalk, RPM packages and, 246
- special characters, escaping, 149
- specialized storage, 5
- Squid Proxy Server, 366
- SSH (Secure Shell), 307
- ssh command, 316
  - remote execution, 320–321
  - remote login, 318–320
- sshd service, 718
  - starting, 317–318
    - at boot, 318
  - status, 317
- Stallman, Richard M., 11
- startup methods
  - init facility, 524–525
  - systemd facility, 525
- statements, echo, 148
- sticky bits, 103, 268–269
- storage
  - cloud, 711, 713
    - configuring, 718–720
    - setup, 714
  - Cockpit, 301–303
  - specialized, 5, 213
  - volume groups, 274
- stream editor (sed), 160
- su command, 168, 175–176
- subscriptions, 20–21
- sudo command, 168, 632–633
- sudo facility, 176–178
- sudoers file, 177, 183
- superuser, 167

- SVID (System V Interface Definition), 10
- swap partitions, 215
- swap space, 4, 274
  - troubleshooting and, 554–555
- SYN (synchronize packet), 666
- syntax, commands, 67–68
- /sys directory, 95
- sysstat package, sar command, 332–333
- System Activity Reporter, 332–333. *See also* sar command
- system administration, 167
  - browser-based tools, 173
  - Cockpit, 168, 169–171
  - commands, 178–179
  - configuration files, 179–185
  - daemon processes
    - apache, 185
    - avahi, 185
    - bin, 185
    - chrony, 185
    - lp, 185
    - news, 186
    - postfix, 185
    - rpc, 186
  - filesystems and, 168
  - graphical tools, 172
  - hardware
    - checking, 187–189
    - loadable modules, 191–193
    - removable, 189–191
  - log files, 183–184
    - rsyslogd facility and, 184–185
  - network interfaces, 169
  - root user, 174
    - via shell, 175–176
  - security, 169
  - servers, 169
  - software installation and, 168
  - system-config\* tools, 171–173
  - systemd journal, 183–184
  - user accounts and, 168
- system administrator, 167
- system logging
  - enabling, 326–331
  - loghost, 330–331
  - logwatch, 331–332
  - message log file, 329
  - rsyslogd daemon, 326–327
- System Monitor, processes
  - ending, 136
  - killing, 136
  - listing, 136–137
  - priorities, 137
  - stopping, 136
- system space
  - disk consumption, 335–336
  - disk usage check, 334–335
  - displaying, 334
- System V, init facility, 524–525
  - troubleshooting, 533
- system-config\* tools, 171
  - system-config-authentication, 172
  - system-config-bind, 172
  - system-config-date, 172
  - system-config-firewall, 172
  - system-config-httpd, 172
  - system-config-kickstart, 173
  - system-config-language, 172
  - system-config-nfs, 172
  - system-config-printer, 172
  - system-config-rootpassword, 172
  - system-config-samba, 172
  - system-config-selinux, 172
  - system-config-services, 172
  - system-config-users, 172
- systemctl command, 381
- systemd, 370–371, 525
  - initialization, 377–384
  - journal, 183–184
  - service units, 378–380
  - services
    - new, 399–401
    - reloading, 390–391
    - restarting, 389–390
    - starting, 389
    - stopping, 389
- SysVinit, backward compatibility, 382–384
- target units, 377, 378, 381–382, 395
- troubleshooting, 538–541
- units, 377
- systems monitoring, 580–581
  - filesystem, 587
  - rootkits, 590–595
  - scanning, 589–590
  - software package verification, 588
  - virus detection, 590–595
- log files, 581



- special commands, 582–583
  - /var/log directory, 581–582
- user accounts, 584
  - bad passwords, 586–587
  - counterfeit accounts and privileges, 584–586
- SysVinit, 370, 372
  - backward compatibility, 382–384
  - runlevels, default, 394–395
- services
  - checking, 385–387
  - new, 396–398
  - persistent, 391–394
  - starting/stopping, 387–391

## T

- tar, 566
  - encryption/decryption, 604
- tarballs, 16, 224–225
- target units, 377, 395
- targeted policy, 638–639
- targets, 369
- TCP Connect port scan, 666, 668
- TCP Wrappers, 313
  - NFS access, 510
- tcsh shell, 61, 65
- TE (type enforcement), 637–638
- telinit command, 374
- telnet, 316
- Terminal emulator, 63–64
- Terminal window, 63
  - GNOME Terminal, 64
  - launching, 64
- terminal windows, 62–63
- test expressions, 154–156
- text
  - adding, vi editor, 115–116
  - changing, vi editor, 117
  - commands, 79
  - copying, vi editor, 117
  - cutting, 159
  - deleting, 160
    - vi editor, 117
  - grep, 159
  - here text, 100
  - moving within, vi editor, 116–117
  - pasting, vi editor, 118
  - searching for, vi editor, 119–120
  - translating, 160
- text editors
  - emacs, 114
  - jed, 114
  - joe, 114
  - kate, 114
  - kedit, 114
  - mcedit, 114
  - nano, 114
  - nedit, 114
  - vi, 113–119
    - arrow keys, 116–117
    - command mode, 115
    - command repeat, 118
    - cursor, 115
    - ex mode, 120
    - exiting, 118–119
    - input mode, 115
    - moving around in file, 119
    - text, 115–120
  - vim, 113–114
- Thompson, Ken, 8
- tilde (~), 97
- /tmp directory, 95
- top command, 134–135
- Torvalds, Linus, 7, 13–14
- touch command, 98–99
- tr command, 160
- training, 21
- translating text, 160
- troubleshooting
  - BIOS (Basic Input Output System), 526–527
  - boot order, 527–528
  - boot up and, 523–524
    - from firmware, 526–528
    - kernel startup, 532–541
      - startup methods, 524–525
  - GRUB 2 boot loader, 530–531
  - GRUB boot loader, 528–530
  - init system, 533
  - memory, 553–554
    - uncovering issues, 554–559
  - networking
    - incoming connections, 550–553
    - outgoing connections, 547–550
  - RAM and, 554–555
  - rc.sysinit, 533–534
  - in rescue mode, 559–561
  - runlevels, 534–538

- troubleshooting (*continued*)
  - software packages, 542–545
    - RPM databases and caches, 545–546
  - systemd initialization, 538–541
- type command, 71

## U

- Ubuntu, 19
  - downloading, 790–791
- Ubuntu Software Center, 225
- UDP port scan, 666
- UEFI (Unified Extensible Firmware Interface), 526–528
- umask value, 108–109
- umount command, 299
- units, systemd, 377
- UNIX, 7–8
  - assembler, 9
  - commercial, Berkeley distribution, 9–10
  - filesystem, 8
  - input/output redirection, 8
  - laboratory, 10–11
  - portability, 8–9
  - print commands, 404
  - printers, remote, 413
  - published interfaces, 10
  - source code, 10
  - USL (UNIX System Laboratories), 10
- unmounting filesystems, 520–521
- untyped variables, 152–153
- updates, Gentoo, 207
- upgrades from scratch, 207
- upstream software providers, 228
- USB drive, 197
  - booting Linux from, 791–792
- user accounts
  - centralizing, 269–270
  - Cockpit, 249–252
  - creating, 249–250
  - defaults, 255–257
  - deleting, 258–259
  - modifying, 257–258
  - security, 566
    - number of users, 567
    - root, access, 567
    - temporary account expiration, 567–568
    - unused, 568–569
  - system administrator and, 168
  - useradd command, 252–255

- user interfaces, 4
- useradd command
  - adding users, 252–255
  - defaults, setting, 255–257
- userdel command, 258–259
- usermod command, 257–258
- username, completion, 75
- USL (UNIX System Laboratories), 10
- /usr directory, 95
- utilities
  - administrative, 4–5
  - backup, 566
  - cracklib, 571
  - iptables, 673–674, 678–688
  - nmap, 665–672
  - programming, 5
- UTS namespace, 695

## V

- /var directory, 95
- variables
  - command output, 149
  - completion, 75
  - environment variables, 81, 82–83
    - adding to shell, 87
    - PATH, 70
  - expanding, 80–81
  - shell, 81
  - shell scripts, 149
  - untyped, 152–153
- Verizon, 9
- Very Secure FTP daemon, 309
- VFAT, 275
- vi editor, 85, 113–114
  - arrow keys, 116–117
  - command mode, 115
  - commands, repeating, 118
  - cursor, 115
  - ex mode, 120
  - exiting, 118–119
  - input mode, 115
  - moving around in file, 119
- text
  - adding, 115–116
  - changing, 117
  - copying, 117
  - deleting, 117
  - moving in, 116–117

- pasting, 118
  - searching for, 119–120
- video, boot options, 210
- vim editor, 113–114
- vimtutor, 177
- virsh command, 711
- virt-install command, 720–721
- virt-manager, 711, 717, 720–724, 722
- virt-manager command, 714
- virtual consoles, 65
- Virtual Machine Manager, 711, 714
  - starting, 722
- virtual memory, 556
- VirtualBox, 209
- virtualization, 5, 709
  - Linux installation, 209
  - viewer, 714
- virt-viewer command, 714
- virus detection, 590
  - intrusion detection, 592–595
  - monitoring for rootkits, 591–592
  - monitoring for viruses, 591
- virus signature, 591
- VMs (virtual machines), 693–694, 709, 713. *See also*
  - hypervisor
  - connections, 722
  - creating, 720–724
  - images, 721
  - installing, 724
  - managing, 724–725
  - migrating, 725–727
  - network bridge and, 721
  - system, viewing, 724–725
- VMware, 209
- VNC installations, boot options, 212
- volume groups, 274
- VPN (virtual private network), 339
- vsftpd, 309, 458–461
  - file permissions, 465
  - installing, 457–458
  - setup, 468–469
- VSZ (virtual set size), 133

## W

- web server, 309
  - Apache HTTPD, 427–428
    - installing, 431
- who am i command, 65
- wildcards, 505
- Winbind, 270
- window manager, 29
- windows
  - graphical, 168
  - terminal windows, 62–63
- Windows (SMB) printer, 414–415
- Windows-based filesystems, 95
- wired networks, 339
- wireless networks, 339
- worker nodes, 710
- Write permissions, 106

## X

- X Window System, 28
  - background, 28
  - clients, 28
  - servers, 28
  - window manager, 29
- Xen, 5, 209, 710
- Xfce, 29
- xinetd.conf file, 183

## Y-Z

- YUM (YellowDog Update Modified)
  - DNF (Dandified YUM), 229
  - packages
    - groups, updating, 239–240
    - installing, 236–237
    - maintenance, 240
    - removing, 236–237
    - searching for, 234–235
    - updating, 238
  - RPM download, 241
  - third-party software repositories, 233
  - yum command, syntax, 229–232
- yum cache, 546
- yum command, 229–241